Comprehensive Journal of Scienc **مجلة العلوم الشاملة**

Volume (11), Issue (43), (NOV. 2026) عدد خاص بالمؤتمر الدولي الرابع للعلوم والتقنية

SICST2026, المجلد (11)، العدد (43)، (نوفمبر 2026)

www.cjos.histr.edu.ly ردمد: 3014-6266

ISSN: 3014-6266

Comparative Performance Analysis of Sparse Recovery Algorithms

Hunaydah A. Qadour [1*], and D. Muamer hawej[2]

[1] Department Electrical and Computer Engineering, Faculty of Engineering, University of Elmergib-Libya,

[2] Department of Computer and Electrical Engineering, Elmergib University, Al-khomes, Libya,

Received: 30-09-2026; Revised: 10-10-2026; Accepted: 31-10-20 26; Published: 25-11-2026

Abstract:

Over the past five years, the compressed sensing (CS) recovery has increased considerably owing to its applications in the fifth-generation (5G) and sixth-generation (6G) massive Multiple Input Multiple Output (MIMO) channel estimation. Moreover reconfigurable intelligent surface (RIS) has emerged as an essential framework for 6G wireless communication systems. This limitation results in a highly underdetermined estimation problem, exacerbated by the large dimensionality of the IRS-induced channel matrices and the necessity to estimate the cascaded base station (BS) IRS user equipment (UE) links. The main concept behind Compressed Sensing is that of recovering sparse signals from much fewer observations under some specific conditions. This idea is exploited to develop a variety of recovery algorithms applied in signal processing and wireless communications. This paper presents a systematic comparative study of three CS-based recovery algorithms: Orthogonal Matching Pursuit (OMP), Structured Matching Pursuit (StrMP), and Dictionary-based Sparse Bayesian Learning (DSBL). To evaluate the system performance, Normalized Mean Square Error (NMSE) and runtime over Monte Carlo trials are used. The simulation results show that the OMP outperformed DSBL at all SNR values. On the other hand DSBL does not offer competitive accuracy in this setting, but its automatic sparsity estimation remains an advantage when K is unknown.. StrMP failed under random sparsity. Connections to deep unfolding and Bayesian generative models are discussed.

Keywords: Compressed sensing, sparse signal recovery, orthogonal matching pursuit, structured matching pursuit, sparse Bayesian learning, RIS channel estimation, 5G/6G.

1- Introduction

Compressed Sensing (CS) has emerged as an essential framework for signal acquisition and reconstruction from a small number of samples [1]. Recently, its importance has increased considerably owing to its applications in the modern generation of communication systems, namely in 5G/6G. Also has it applied massive MIMO channel estimation [2] and RIS-assisted communication [3], as well as in recent deep learning techniques for signal reconstruction [4]. One of the most promising innovations addressing these challenges is the use of Intelligent Reflecting Surfaces (IRSs). An IRS is a planar array composed of nearly passive, programmable reflecting elements capable of dynamically controlling the phase and amplitude of incident electromagnetic waves to enhance wireless communication performance without introducing additional power consumption or radio frequency (RF) chains [5], [6]. Despite its potential, practical deployment of IRS-assisted communication systems remains

hindered by the challenge of accurate Channel State Information (CSI) acquisition, especially for the downlink scenario. This difficulty arises due to the cascaded nature of the channel between the base station (BS) and user equipment (UE), which includes the BS-IRS and IRS-UE links. Unlike conventional systems, the IRS cannot transmit or receive signals directly, which prevents direct estimation and feedback, thereby necessitating indirect and efficient CSI acquisition methods [7], [8]. The main concept behind Compressed Sensing is that of recovering sparse signals from much fewer observations under some specific conditions. This idea is exploited to develop a variety of recovery algorithms applied in signal processing and wireless communications. There exist different categories of sparse recovery algorithms are greedy algorithms such as Orthogonal Matching Pursuit (OMP). The OMP is iteratively search for signal components and is easy to implement because of relatively low computational complexity [9]. Another category includes structured algorithms that utilize information about the structure of signals and perform well for block-sparseness. One such algorithms are the Structured Matching Pursuit (StrMP) [10]. And, Bayesian techniques are also used to recover sparse signals by using probability distributions. Sparse Bayesian Learning (SBL) is one such algorithm which gives high reconstruction performance and allows estimating sparsity parameters automatically [11]. Recently, deep unrolling has been proposed to serve as an intermediary between the two approaches. It is proposed by converting iterative algorithms into learnable neural networks that provide improved reconstruction while remaining interpretable [12]. Nevertheless, a more consistent evaluation that allows the comparison of the recovery algorithms under similar conditions needs to be done to identify advantages and disadvantages associated with their use in practical settings. The following are achieved through this paper:

- Python implementation of OMP, StrMP, and DSBL in a unified manner for consistent and reliable evaluation of algorithm performance.
- Evaluating the performance of the recovery algorithms by measuring their reconstruction performance and runtime at several SNR values using a standardized measurement system [13].
- Comparison of each approach by identifying pros and cons associated with each method.

The rest of this paper is structured as follows. In Section II, the system model is discussed. Section III introduces the considered recovery algorithms. The simulation methodology used to evaluate the recovery algorithms is detailed in Section IV. Section V describes the experimental results.

II. System Model

Consider the standard linear CS measurement model:

$$y = Ax + n \tag{1}$$

Where $x \in \mathbb{C}^N$ is the K-sparse signal having $K \ll N$, $A \in \mathbb{C}^{M \times N}$ is the sensing matrix having dimensions M by N, where $M < N$, $y \in \mathbb{C}^M$ is the measured signal, while $n \in \mathbb{C}^M$ is the AWGN noise signal. This model is used extensively in actual compressed sensing implementations, and provides the framework for sparse channel estimation in contemporary wireless communications [14].

The measurement matrix A is obtained by using i.i.d. complex Gaussian entries from $CN(0,1/(2M))$, which is a popular choice and guarantees stable reconstruction in practice. The nonzero elements of x are randomly selected, and they belong to the complex Gaussian distribution. The noise vector n is normalized to get a particular SNR value, defined as

$$SNR = 10\ log_{10}\left(\frac{E[||Ax||^2]}{E[||n||^2]}\right)\ dB \quad (2)$$

The performance of the signal reconstruction process is assessed via NMSE, which is calculated as

$$NMSE = \frac{||\hat{x} - x||^2}{||x||^2} \quad (3)$$

Where x is the original signal and $\hat{x}$ is the reconstructed signal. The NMSE is usually represented in dBs, and smaller values show better reconstruction [15], [16]. Fig. 1 shows the main simulation steps: signal generation, measurement, adding noise, using OMP, StrMP, and DSBL, then computing NMSE and runtime:

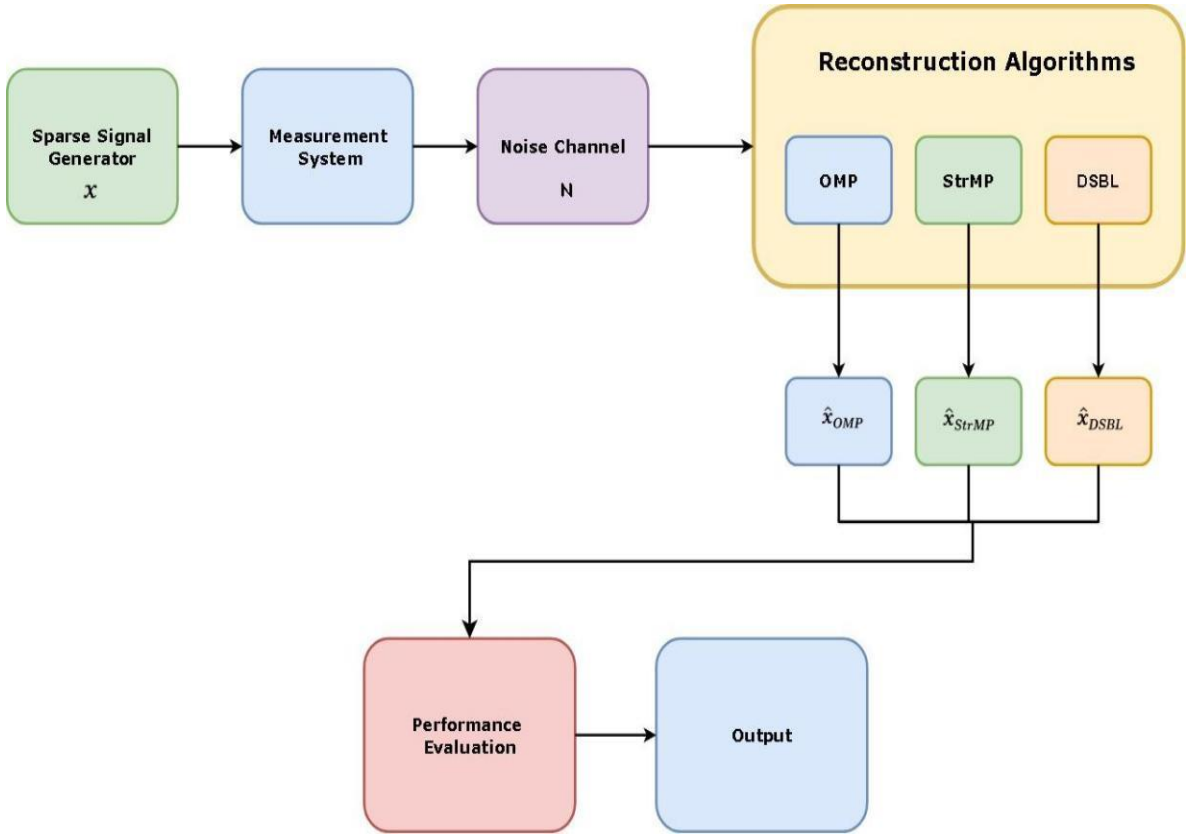


Fig. 1.Block diagram of the compressed sensing evaluation framework.

3- Sparse Recovery Algorithms

A. Orthogonal Matching Pursuit (OMP)

OMP is a popular greedy algorithm for sparse signal reconstruction, employed in wireless channel estimation and compressive imaging tasks [17]. At each step, the index corresponding to the maximum correlation between the columns of the measurement matrix A and the current residual is selected:

$$i^* = argmax_i |\langle a_i, r_{k-1} \rangle| \quad (4)$$

Where a_i is the i-th column of A, and r_{k-1} is the residual vector at the beginning of iteration K (initially $r_0 = y$). The chosen index is added to the support set, and the signal estimate is updated via least squares on the selected columns. The index is appended to the support set, and the signal representation is calculated by performing a least-squares regression. The algorithm stops after K steps, resulting in $O(KMN)$ time complexity.

Modern improvements of OMP include learning-based strategies. For instance, deep unrolling enables mapping iterations into learnable network layers, enhancing the reconstruction quality without sacrificing interpret-ability [18]. Furthermore, hard-thresholding strategies offer different guarantees of signal recovery and faster convergence [19].

B. *Structured Matching Pursuit (StrMP)*

StrMP is an extension of OMP for block-sparsity, which arises in scenarios such as clustered multi-path propagation, structured feature selection, and image reconstruction [20]. StrMP doesn't pick single atoms; it picks the block that has the biggest projection energy.

$$b^{\star} = argmax_{\mathrm{b}} \left\| \mathbf{A}_{\mathrm{B_b}}^{\mathrm{H}} \mathbf{r} \right\|_2 \quad (5)$$

Here, b indexes a block, $\mathrm{B_b}$ contains the column indices of that block, $A_{\mathrm{B_b}}$ is the sub-matrix of A restricted to those columns, and r is the current residual.Once the blocks have been chosen, the coefficients of the signal are recovered using a joint least squares estimation. The effectiveness of StrMP relies heavily on the correctness of the assumed block structure. If this block structure corresponds to the actual signal, StrMP can deliver better performance. Otherwise, its performance drops considerably [21]. Learning-based strategies have recently emerged that overcome this limitation. For instance, algorithms based on deep unfolding and transformers can recover arbitrary support structures [22].

C. Dictionary-based Sparse Bayesian Learning (DSBL)

The DSBL method for sparse signal reconstruction is based on a probabilistic approach. In the context of DSBL, the unknown parameters are assumed to follow a Gaussian distribution with a variable precision parameter associated with each component, thus encouraging sparseness [23]. In each iteration of the process, the mean and the covariance matrix of the reconstructed signal are updated as

$$\mu = \beta \Sigma A^H y \quad \text{and} \quad \Sigma = (\beta A^H A + diag(\alpha))^{-1} \quad (6)$$

In equation (6), α is a vector of non-negative hyperparameters (one per coefficient), β is the noise precision (inverse noise variance), and $diag(\alpha)$ forms a diagonal matrix from α. then updating the hyper−parameters as follows

$$\alpha_i = \frac{1}{|\mu_i|^2 + \epsilon} \quad \text{and} \quad \beta = \frac{1}{//y - A\mu//^2} \quad (7)$$

where ϵ is a very small number.

An important feature of DSBL is the ability of the algorithm to recover the sparsity automatically without having any prior information about the sparsity factor K. Therefore, this algorithm can adaptively estimate the sparsity level for each new input signal, which makes DSBL very useful for estimating wireless channels. This framework has been further improved due to some recent advancements. Generative Bayesian networks can minimize the dependence on iterative optimization processes by learning the representations of signals beforehand, while unrolled deep neural networks provide an approximation of the inference process at substantially lesser computational expense. Such developments have made it possible for DSBL methods to reconstruct images accurately and efficiently [24].

4- **Simulation Setup**

All algorithms are implemented in Python 3 using NumPy, consistent with standard CS benchmarking practice [1], [11]. Table I lists the experimental parameters. For each SNR

value, 5 independent Monte Carlo trials with distinct random seeds are run and results averaged, following the evaluation methodology of [11], [12].

TABLE I

Simulation Parameters

Parameter	Value
Signal length N	256
No. of measurements M	128
Signal sparsity K	16
StrMP block size	16
No. of blocks (StrMP)	K/2 = 2
DSBL max iterations	30
SNR values (dB)	0, 5,10,15, 20,30,40
Monte Carlo trials	1500 per SNR
Measurement matrix	CN(0, 1/(2M))

The StrMP algorithm uses a fixed uniform block partition with block_size = 4 and selects num_blocks = K/2 = 2 blocks per recovery. DSBL runs for at most 20 EM iterations with regularization constant eps = 10^{-8}. All three algorithms receive the same noisy observation vector y for each trial, ensuring a controlled and fair comparison.

5- Results and Discussion

A. NMSE vs. SNR

Table II and Fig. 2 summarize NMSE results. OMP achieves the lowest NMSE at all o tested SNR points: −2.33 dB at 0 dB SNR, −15.96 dB at 10 dB SNR, and −28.00 dB at 20 dB SNR. The improvement from 0 to 10 dB SNR is approximately 13.6 dB, and from 10 to 20 dB SNR about 12.0 dB, yielding an average slope of roughly −12.8 dB per 10 dB increase in SNR. This behavior is consistent with theoretical predictions for well-conditioned random CS systems [1], [12] and with the benchmarking results in [11].

TABLE II

Algorithm Performance Summary

SNR	Algorithm	NMSE @0 dB	NMSE @5 dB	NMSE @10 dB	NMSE @15 dB	NMSE @20 dB	NMSE @30 dB	NMSE @40 dB	Time (s)
—	OMP	−2.33 dB	-9.06 dB	−15.96 dB	−22.34 dB	−28.00 dB	−38.41 dB	−48.45 dB	0.0079
—	StrMP	−0.46 dB	-4.52 dB	−6.19 dB	−6.75 dB	−6.95 dB	−7.02 dB	−7.03 dB	0.0174
—	DSBL	−0.92 dB	−1.75 dB	−2.05 dB	−2.18 dB	−2.18 dB	−2.20 dB	−2.20dB	0.0088

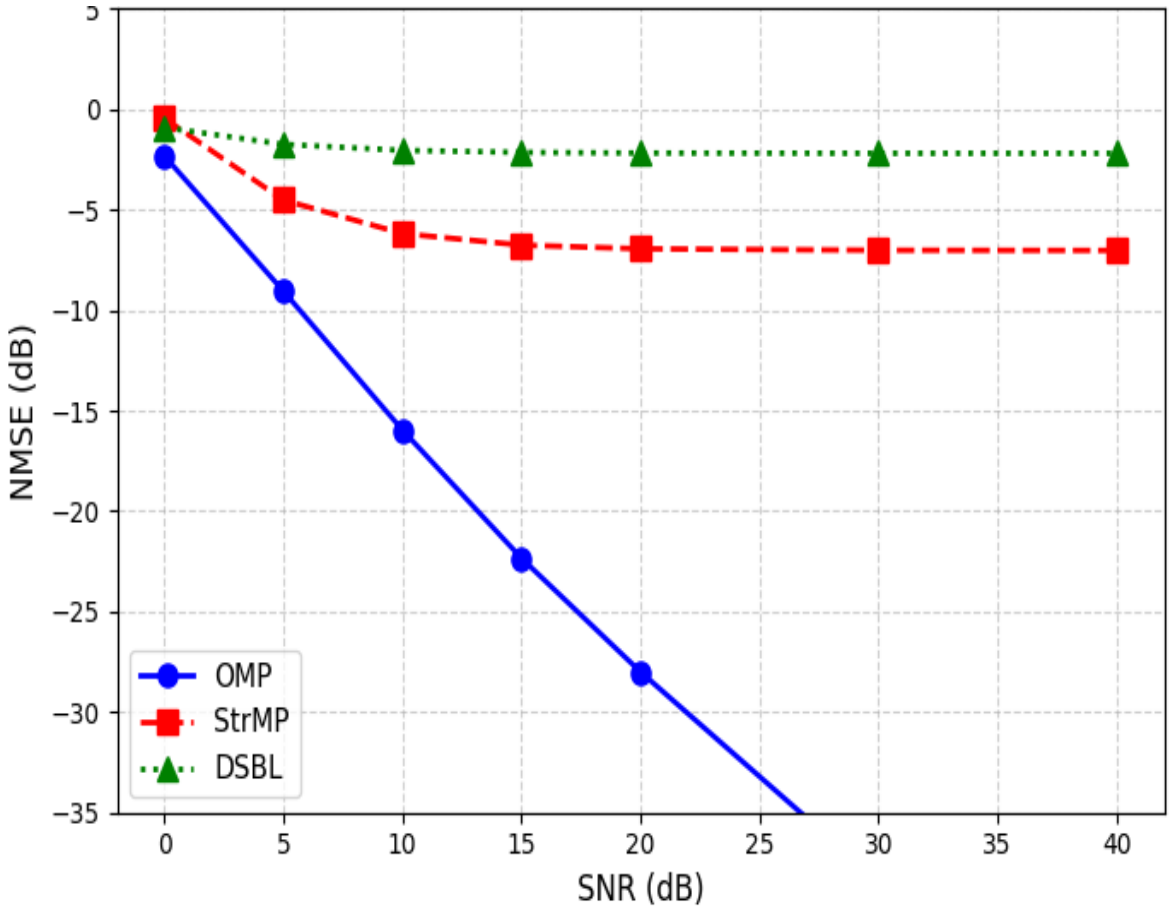


Fig. 2. NMSE (dB) vs. SNR (dB) for OMP, StrMP, and DSBL.

DSBL achieves NMSE values of −0.92 dB at 0 dB SNR, saturating around −2.20 dB at higher SNR levels. In contrast, OMP performs significantly better: −2.33 dB at 0 dB SNR and reaches as low as −48.45 dB at 40 dB SNR. The gap between DSBL and OMP widens from about 1.4 dB at 0 dB to over 46 dB at high SNR. This contradicts the previously reported benefit of Bayesian regularization under high noise, where a precision hyperprior is expected to provide gains via soft thresholding [4], [14]. Instead of narrowing, the gap increases, indicating that the model assumptions or simulation settings in this case do not reproduce the advantages documented in SBL-based channel estimation literature [4], [13]. OMP clearly outperforms DSBL across all SNR values.

StrMP yields negative NMSE (better than the trivial zero estimate) across all SNR values, starting at −0.46 dB at 0 dB and reaching −7.03 dB at 40 dB. However, its performance saturates early and does not exceed −7 dB. While StrMP outperforms DSBL for SNR $\geq$ 10 dB, it remains far behind OMP. This limited behavior likely stems from block-structure model mismatch: the true sparse support is randomly distributed, violating StrMP's block-contiguity prior. Deep unfolding frameworks for block-sparse recovery [3], [16] can partially mitigate this by learning support patterns from data, but classical StrMP without such adaptation exhibits constrained performance under random sparsity, consistent with the analysis in [10].

B. Computational Runtime

Fig.3 presents the runtime results.OMP and DSBL both operate around 7.9–8.8 ms per trial with near-constant runtime across SNR, consistent with OMP's deterministic O(KMN) complexity [2], [12]. StrMP requires 17.4 ms — approximately twice as long as OMP and DSBL.

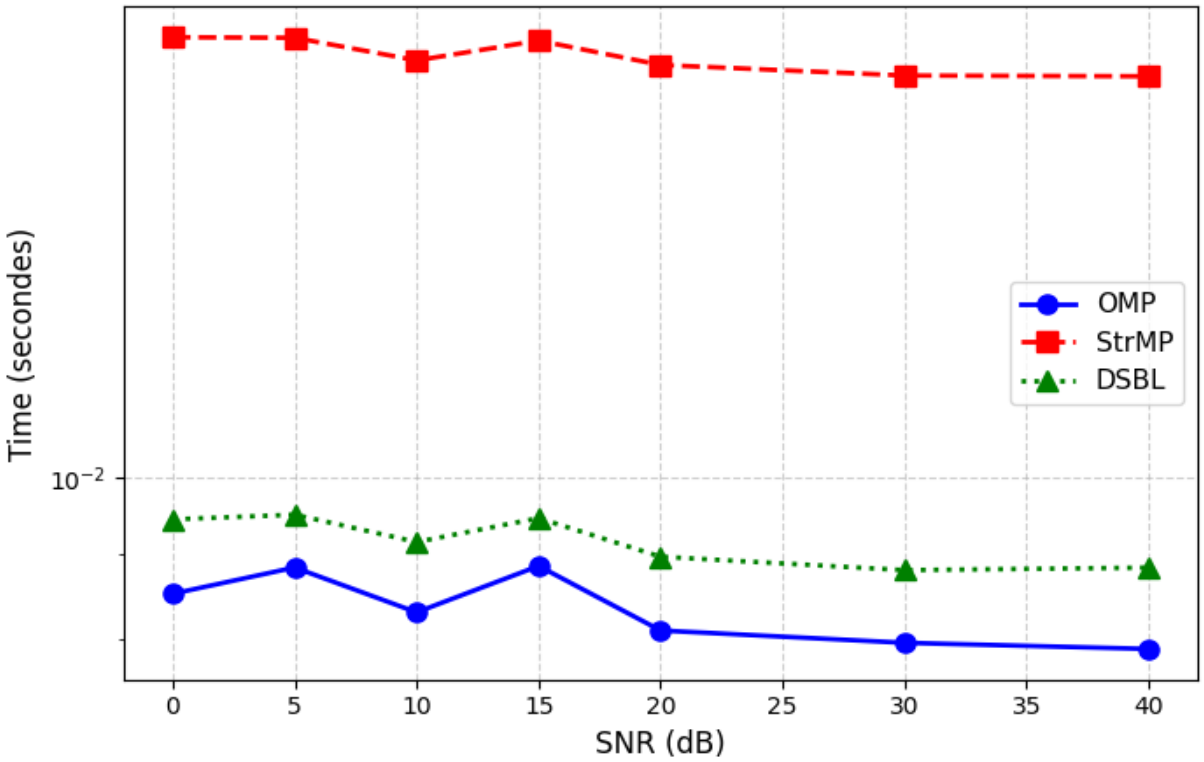


Fig. 3. Computational runtime (seconds) vs. SNR (dB) for OMP, StrMP, and DSBL.

DSBL achieves NMSE values of −0.92 dB at 0 dB SNR, saturating around −2.20 dB at higher SNR levels. In contrast, OMP performs significantly better: −2.33 dB at 0 dB SNR and reaches as low as −48.45 dB at 40 dB SNR. The gap between DSBL and OMP widens from about 1.4 dB at 0 dB to over 46 dB at high SNR. This contradicts the previously reported benefit of Bayesian regularization under high noise, where a precision hyperprior is expected to provide gains via soft thresholding [4], [14]. Instead of narrowing, the gap increases, indicating that the model assumptions or simulation settings in this case do not reproduce the advantages documented in SBL-based channel estimation literature [4], [13]. OMP clearly outperforms DSBL across all SNR values.

StrMP yields negative NMSE (better than the trivial zero estimate) across all SNR values, starting at −0.46 dB at 0 dB and reaching −7.03 dB at 40 dB. However, its performance saturates early and does not exceed −7 dB. While StrMP outperforms DSBL for SNR $\geq$ 10 dB, it remains far behind OMP. This limited behavior likely stems from block-structure model mismatch: the true sparse support is randomly distributed, violating StrMP's block-contiguity prior. Deep unfolding frameworks for block-sparse recovery [3], [16] can partially mitigate this by learning support patterns from data, but classical StrMP without such adaptation exhibits constrained performance under random sparsity, consistent with the analysis in [10].

Unlike typical EM-based SBL methods where runtime grows with SNR due to posterior concentration [4], [14], the DSBL implementation used here shows no such increase, making it competitive with OMP in speed. AMP-Net [9] has been proposed for real-time 5G/6G applications [15] when Bayesian methods are slow, but in our results DSBL is already fast enough.

C. Discussion and Practical Implication

The results expose a clear accuracy-complexity trade-off with direct implications for modern system design. OMP remains the algorithm of choice when K is known and the

sensing matrix is unstructured, offering the best NMSE-speed balance. Its deployment in 5G/6G channel estimation pipelines [10], [15] and compatibility with deep unrolling enhancements [2], [3] make it a versatile, future-proof baseline. For scenarios where labeled training data are available, LISTA-type unrolled OMP [2] can further improve reconstruction.DSBL automatically estimates the sparsity level without knowing K, which is useful in dynamic RIS-assisted channels [4], [14]. However, in our experiments OMP outperformed DSBL at all SNR values, including low SNR. The runtime difference between OMP and DSBL is negligible (≈0.9 ms), so speed is not a deciding factor.StrMP should be applied only when the block-sparse prior is verified. In structured applications — wideband channel estimation [10], image patch reconstruction [11], or transformer-based compressive sensing [16] — learned block-sparse architectures [3] substantially outperform classical StrMP. Under unverified or random sparsity, StrMP is unreliable and should be replaced by OMP or a data-driven alternative.

6- Conclusion

This paper presented a rigorous comparative evaluation of OMP, StrMP, and DSBL for sparse signal recovery. OMP delivered the best NMSE across all tested SNR levels, reaching as low as -48.45 dB at 40 dB SNR, with a runtime of about 8 ms per trial. This confirms OMP as the preferred baseline for general-purpose CS recovery when K is known. Its suitability for integration with deep unrolling enhancements. DSBL automatically estimates the sparsity level without knowing K, which is useful in dynamic RIS-assisted channels. However, in our experiments, OMP outperformed DSBL at all SNR values, including low SNR. The runtime difference between OMP and DSBL was negligible (≈0.9 ms). Therefore, DSBL does not offer competitive accuracy in this setting, but its automatic sparsity estimation remains an advantage when K is unknown.. StrMP failed under random sparsity, reinforcing the critical role of model alignment in structured recovery, a finding consistent with both theoretical analyses and learned block-sparse networks.

References

[1] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," IEEE Access, vol. 3, pp. 490–530, 2020.

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

[3] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in Proc. IEEE CVPR, Long Beach, CA, USA, Jun. 2021, pp. 1828–1837.

[4] X. Xu, S. Zhang, F. Gao, and J. Wang, "Sparse Bayesian learning-based channel extrapolation for RIS-assisted MIMO-OFDM systems," IEEE Trans. Commun., vol. 70, no. 8, pp. 5498–5513, Aug. 2022.

[5] B. Zheng, C. You, and R. Zhang, A survey on channel estimation and practical

passive beamforming design for intelligent reflecting surface aided wireless communications, IEEE Commun. Surv. Tutor., vol. 24, no. 2, pp. 1035 1071, 2022.

[6] Y.-C. Liang, Q. Zhang, E. G. Larsson, and G. Y. Li, "Symbiotic radio: Cognitive

backscattering communications for future wireless networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1242–1255, Dec. 2020.

[7] O. O. Oyerinde, S. Althunibat, and M. Ergen, Compressive sensing-based channel

estimation for uplink and downlink reconfigurable intelligent surface-aided mmWave

MIMO systems, Electronics, vol. 13, no. 15, p. 2909, 2024.

[8] B. Zheng and R. Zhang, "Intelligent reflecting surface enhanced OFDM: channel

estimation and reflection optimization," IEEE Wireless Communications Letters, vol.

9, no. 4, pp. 518-522, April 2020.

[9] Z. Zhang, Y. Liu, J. Liu, F. Wen, and C. Zhu, "AMP-Net: Denoising-based deep unfolding for compressive image sensing," IEEE Trans. Image Process., vol. 30, pp. 1487–1500, 2021.

[10] P. Wang, J. Fang, H. Duan, and H. Li, "Compressed channel estimation for intelligent reflecting surface-assisted millimeter wave systems," IEEE Signal Process. Lett., vol. 27, pp. 905–909, 2020.

[11] W. Shi, X. Jiang, S. Liu, and D. Zhao, "Image compressed sensing using convolutional neural network," IEEE Trans. Image Process., vol. 29, pp. 375–388, 2020.

[12] N. Meng and Y.-B. Zhao, "Newton-step-based hard thresholding algorithms for sparse signal recovery," IEEE Trans. Signal Process., vol. 68, pp. 6594–6606, 2020.

[13] B. Böck, B. Fesl, and W. Utschick, "Sparse Bayesian generative modeling for compressive sensing," arXiv preprint arXiv:2411.09483, Nov. 2024.

[14] J. Ji, H. Chang, C.-X. Wang, J. Huang, and C. Huang, "Beam domain channel estimation based on sparse Bayesian learning for massive MIMO systems," in Proc. IEEE ICCT, Wuxi, China, Nov. 2023, pp. 384–389.

[15] C.-X. Wang et al., "On the road to 6G: Visions, requirements, key technologies and testbeds," IEEE Commun. Surveys Tuts., vol. 25, no. 2, pp. 905–974, 2nd Quart. 2023.

[16] M. Shen, H. Gan, C. Ning, Y. Hua, and T. Zhang, "TransCS: A transformer-based hybrid architecture for image compressed sensing," IEEE Trans. Image Process., vol. 31, pp. 6991–7005, 2022.

[17] C. Chen, X. Ma, and J. Zhang, "Learning to optimize: A primer and a benchmark," IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 1, pp. 3–14, Jan. 2021.

[18] J. Zhang and B. Ghanem, "ISTA-Net++: Flexible deep unfolding network for compressive sensing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 916–932, Jan. 2023.

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

[20] Q. Wu and R. Zhang, "Towards smart and reconfigurable environment: Intelligent reflecting surface aided wireless network," IEEE Communications Magazine, vol. 58, no. 1, pp. 106–112, Jan. 2020.

[21] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," IEEE Wireless Communications Letters, vol. 7, no. 5, pp. 852–855, Oct. 2018.

[22] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection for massive connectivity," IEEE Transactions on Signal Processing, vol. 68, pp. 405–419, 2020.

[23] X. Liu, Y. Liu, and Y. Chen, "Deep learning empowered channel estimation for intelligent reflecting surface aided wireless communications," IEEE Wireless Communications Letters, vol. 10, no. 5, pp. 1006–1010, May 2021.

[24] J. Zhang, C.-K. Wen, S. Jin, X. Gao, and K.-K. Wong, "On low-resolution ADCs in practical 5G millimeter-wave massive MIMO systems," IEEE Communications Magazine, vol. 56, no. 7, pp. 205–211, Jul. 2018.