Enhanced Beamspace Channel Recovery in mmWave MIMO Using Deep Neural Network

V.Saraswathi[1], Vatsala Anand[2], Yaddanapudi Venkata Bhaskara Lakshmi[3], Samana Vinaya Kumar[4],Vijaya Babu Burra[5], B.Rajani[6], U.S.B.K.MAHALAXMI[7], Suneetha Jalli[8], Dr V Vijayasri Bolisetty[9], Dr Sarala Patchala[10]

[1]ECE department, Rajeev Gandhi Memorial College of engineering and technology. Nandyal, Andhrapradesh, India
[2] Department of Computer Science and Engineering, Akal University, Talwandi Sabo, Bathinda, Punjab, India
[3]Associate professor, Dept Of ECE, BABA Institute of Technology & Sciences, PM Palem, Vishakapatnam, Andhara Pradesh, India
[4]Department of Electronics and Communication Engineering, Aditya University, Surampalem, Andhra Pradesh, India
[5]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India
[6]Associate professor, Department of Electrical and Electronics Engineering,Aditya university, Surampalem,A.P, India
[7]Department of Electronics and Communication Engineering, Aditya University, Surampalem, Andhra Pradesh, India
[8]KKR & KSR Institute of Technology and Sciences, Guntur, Andhra Pradesh, India
[9]Associate Professor, Aditya University, Surampalem,Andhra Pradesh, India
[10]Associate Professor, KKR & KSR Institute of Technology and Sciences, Guntur, Andhra Pradesh, India

Abstract

Millimeter-wave (mmWave) massive MIMO systems use many antennas. These systems offer high data rates. But using many radio frequency (RF) chains increases cost and power use. To solve this, lens antenna arrays are used. Energy is focused, allowing the use of fewer RF chains. However, this creates a new challenge. With fewer RF chains, it is hard to estimate the wireless channel. Accurate channel estimation is needed for good system performance. In beamspace, the channel is sparse. This shows that only a few values are large. The rest are close to zero. Because of this, the problem is seen as sparse signal recovery. AMP (Approximate Message Passing) is one popular algorithm used for this. A better version named LAMP (Learned AMP) uses deep learning. But it still does not give the best results. This paper proposes a new method GM-LAMP. It improves the channel estimation accuracy. It uses prior knowledge about the channel. It assumes that the beamspace channel follows a Gaussian mixture distribution. First, a new shrinkage function is created based on this distribution. Then, the original function in the LAMP network is replaced with the new one. As a result, a better deep learning model is developed. The final GM-LAMP network estimates the beamspace channel more precisely. It works well with both theoretical models and real-world data. Simulations show that GM-LAMP performs better than earlier methods. This approach combines math knowledge and deep learning. It shows that using prior information helps deep networks make smarter predictions. The proposed method offers better accuracy and is useful for future mmWave systems.

Keywords: MIMO, mmwave, Channel Recovery, Beam space, deep neural network

Received on 11 September 2025, accepted on 10 November 2025, published on 26 November 2025

doi: 10.4108/eetiot.10247

*Corresponding author. Email: saralajntuk@gmail.com

1. Introduction

mmWave technology is a key part of 5G and future wireless systems[1]. It works at very high frequencies, which allows it to provide very fast data rates. However, mmWave signals face many challenges. These systems experience high path loss, causing signals to weaken quickly as it travels[2]. To solve this, systems use many antennas at the base station. This technique is named as massive MIMO (Multiple-Input Multiple-Output). It helps increase signal strength and improve performance. But using many antennas creates new problems. Each antenna needs a RF chain[3]. These chains convert digital signals into wireless ones and vice versa. Having one RF chain for every antenna becomes expensive and uses a lot of power. This is not practical in real systems. To fix this, researchers use a lens antenna array. This lens focuses signals from different directions onto different antennas. The main approach is to turn a high-dimensional spatial channel into a beamspace channel [4]. In this form, the signal becomes sparse. Most parts of the signal are close to zero, and only a few parts are large. This property helps reduce the number of RF chains needed. To use the lens array effectively, the system needs knowledge of the channel. This process is referred to as channel estimation. Accurate channel information is needed for beam selection and data transmission[5]. However, with fewer RF chains and many antennas, estimating the channel becomes very hard. This is the point at which smart algorithms are needed.
Researchers tried different ways to solve this problem. One simple way is to send pilot signals from users to the base station. The base station then measures the response and tries to estimate the channel[6]. Some methods use beam training. The system checks every possible direction to find the best beam. But this needs many pilot signals, which is not efficient. Other solutions apply concepts from compressive sensing (CS). These methods work well for sparse signals[7]. One popular algorithm is Orthogonal Matching Pursuit (OMP). It picks the strongest parts of the signal first. Then it tries to reconstruct the full signal from those parts. There are matrix completion methods and two-stage estimation techniques[8]. These use both sparse recovery and structure in the signal. But all of these methods still face problems. Performance drops in noisy situations (low SNR) and the approaches require many calculations. To overcome these limits, researchers turned to deep learning (DL). DL is good at learning from data. It has worked well in fields like image and speech recognition. Now, it is being used in wireless communication. DL learn patterns in large data and find good solutions without needing exact math models[9]. To solve this, some researchers combine deep learning with traditional algorithms. The structure of existing methods is taken and improved using learning. One such method is LAMP. It forms a deep network based on the AMP algorithm[10]. Each layer in the network is like one step in the AMP iteration. The network learns better shrinkage parameters for each layer[11]. This helps improve performance.

This paper introduces a new solution based on this approach. It introduces the GM-LAMP network. The GM stands for Gaussian Mixture. This method adds prior knowledge to the LAMP network. It uses a new shrinkage function based on the Gaussian Mixture distribution[12]. This function is added to the deep network layers. The GM-LAMP network has the same structure as LAMP but uses this new function. The paper finishes by linking the estimation quality to beam selection. A better channel estimate leads to better beam choices. This improves data rates. Even with some estimation error, GM-LAMP still achieves good performance[13]. The key contributions of the paper are:

- ❖ To develop a novel method named GM-LAMP for channel estimation in mmWave MIMO systems. To integrate deep learning with prior knowledge of the wireless channel.
- ❖ To design a new Gaussian Mixture-based shrinkage function within the LAMP framework. To train the GM-LAMP network layer by layer for better accuracy and reduced overfitting.
- ❖ To evaluate the method on both simulated and real-world datasets. To demonstrate that GM-LAMP achieves higher accuracy and efficiency than existing approaches.
- ❖ To analyze the impact of channel estimation quality on system-level performance, including beam selection.

This paper offers a new way to improve beamspace channel estimation. It combines classic algorithms with deep learning and uses prior knowledge to get better results[14]. This makes it useful for next-generation wireless systems.

2. Related Work

This section discusses previous studies on mmWave MIMO systems and channel estimation. It explains past efforts and points out problems that still exist. The purpose is to highlight the novelty of this work. Millimeter-wave systems play a key role in 5G and future wireless networks. These systems use many antennas, which help in beamforming and spatial multiplexing. To address the estimation challenge, earlier works used compressed sensing (CS) methods. These are useful for sparse signals. For example, the work in [15] introduced a CS-based method for channel estimation. The mmWave channel was shown to be sparse in the angular domain. So, compressed sensing (CS) was applied to find a low-dimensional version of the channel. Another approach is OMP, used in [16]. This is a greedy algorithm. It selects the strongest signal components first and then attempts to reconstruct the full signal. OMP works well, but it is slow and less accurate in noisy environments. Later, improved CS techniques were explored. One such method is Basis Pursuit (BP), which uses convex optimization. But it takes a long time to compute. Another method is the AM) algorithm. This is faster and works well with Gaussian signals. Many papers widely explore and improve the AMP method.

For example, [17] introduced the Generalized AMP (GAMP) algorithm. It works for a wider set of signals. Then, the LAMP method was proposed in [18]. LAMP uses deep learning. It forms a neural network that follows the steps of the AMP algorithm. Each layer of the network acts like one step in AMP. This network learns better parameters during training. As a result, LAMP gives better performance than standard AMP. DL has become popular for wireless communication. Many works use DL for channel estimation. For instance, [19] used an autoencoder to learn the process of recovering the channel from pilot signals. The method is fully data-driven. It learns from many training samples. Other works, including [20], combined deep learning with expert knowledge. The method used structured channel information and added learning to improve results. This hybrid approach gives better accuracy. The paper in [21] followed this approach. It used DL to improve direction-of-arrival (DOA) estimation. This is important for beam selection. However, many DL methods treat the neural network as a black box. The reason why the network works is not explained. This makes it hard to trust and improve these models.

To solve this, researchers proposed model-driven deep learning. This approach was applied in [22], using steps from classic algorithms to design neural networks. Such networks are easier to understand and control. Fewer training samples are required. One example is LAMP [18], based on the AMP algorithm. LAMP shows success in many applications, including channel estimation and image recovery. But even LAMP has some limits. It does not use prior knowledge about the channel. In real systems, the values in the beamspace channel typically follow a specific distribution. Some works attempt to include prior knowledge. For instance, [23] used Bayesian methods for signal recovery. These methods assume a prior distribution and update the estimate using observed data. However, training is difficult, and performance is slow in practice. Other works like [24] designed networks with built-in structure. Still, detailed statistical models like Gaussian Mixture were not used. This paper takes the next step. It uses the GM model to create a new shrinkage function. This function is placed inside a LAMP-like network. The result is termed as GM-LAMP. The GM-LAMP network is model-driven. It is easy to understand and uses prior knowledge. It learns better parameters through training. The results show that it performs better than older methods. It is fast and works well in real-world settings.

3. System Model

In this section, the paper presents the system model used for mmWave massive MIMO communication. The system focuses on beamspace representation, which is important for reducing hardware complexity while maintaining high signal quality. The section first explains the use of lens antenna arrays and then develops the mathematical channel model. It finally discusses the formation of the received signal and the definition of the estimation problem. The discussion starts with a single-cell mmWave massive MIMO uplink communication system. In this system, the BS is equipped with N antennas. The user equipment (UE) is assumed to use a single antenna. This configuration is commonly used in practical mmWave systems to support simple user devices while achieving high data rates through base station beamforming.

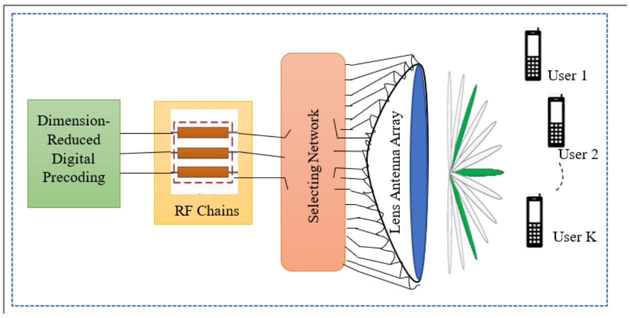

Figure 1: mmWave Massive MIMO with Lens Antenna Array

The Figure 1 shows a communication system using a lens antenna array to serve multiple users. On the left side, there is a block labeled Dimension-Reduced Digital Precoding, which handles digital signal processing with reduced complexity. The output of this block connects to RF Chains, which generate and convert radio signals for transmission. These RF chains pass signals to a Selecting Network. This network selects which paths to activate, controlling the way signals reach the antenna elements. After that, the signals go to the Lens Antenna Array, a special antenna structure that focuses and directs the signals into narrow beams. The lens antenna array directs beams toward different users shown on the right side of the diagram (User 1, User 2, ..., User K). Each user receives a focused signal from a selected beam. The green lines indicate the beams that are actively sending data to the users. This setup improves signal quality and reduces interference. The purpose is to send the right signal to each user using fewer resources and lower processing costs. This type of system is useful in millimeter-wave or massive MIMO communication, requiring efficient beamforming and user separation.

To lower the hardware cost and energy consumption, the base station does not use one RF chain for each antenna. Instead, it uses a smaller number of RF chains denoted as N_{RF}, here $N_{RF} \ll N$. To enable beam selection and simplify processing, a lens antenna array is installed at the base station. This lens maps the spatial domain into beamspace, concentrating the signal energy in a few dominant beams. As a result, beamspace signals are sparse and the estimation process becomes more efficient. The mmWave wireless channel has a sparse structure in the angular domain. In other words, most of the channel power is concentrated in a small number of directions. This feature is captured using a geometric Saleh-Valenzuela model. The channel vector $h \in C^N$, which represents the uplink channel from the user to the base station, is modeled as a superposition of L propagation paths. Each path has its own complex gain α_ℓ, angle of arrival (AoA) θ_ℓ and steering vector $a(\theta_\ell)$. The channel expression is given by:

$$h = \sqrt{\frac{N}{L}} \sum_{\ell=1}^{L} \alpha_\ell \, a(\theta_\ell) \tag{1}$$

Here, the scalar factor $\sqrt{N/L}$ is used for power normalization. The steering vector $a(\theta)$ describes the phase shifts introduced by the antenna array for a given angle θ. If the base station uses a uniform linear array (ULA), the steering vector is expressed as:

$$a(\theta) = \frac{1}{\sqrt{N}} \left[1, e^{j2\pi \frac{d}{\lambda} \sin(\theta)}, \ldots, e^{j2\pi \frac{d}{\lambda} (N-1)\sin(\theta)} \right]^T \tag{2}$$

In the above formula, d represents the spacing between adjacent antennas. λ is the wavelength of the carrier frequency. The steering vector has a unit norm, which keeps the power stable in the channel model. To perform beamspace transformation, the base station uses a lens antenna array, which is mathematically modeled using a unitary discrete Fourier transform (DFT) matrix $U \in C^{N \times N}$. This transformation helps project the spatial domain channel into the beam domain. The resulting beamspace channel vector is given by:

$$\tilde{h} = U^H h \tag{3}$$

Since the angular domain channel is sparse, the beamspace channel $\tilde{h}$ is sparse. Most of its entries are close to zero and only a few entries carry strong energy. This sparsity is useful for efficient channel estimation. During uplink training, the user sends pilot signals to the base station. The base station uses a beam selector to connect some of the available beams to the limited RF chains. The measurement at the base station after beam selection is represented as:

$$y = WU^H h + n = W\tilde{h} + n \tag{4}$$

Here, $W \in C^{M \times N}$ is a beam selection matrix. Here, M is the number of measurements or pilots. Each row of W has only one non-zero element with unit magnitude, indicating that only one beam is connected to each RF chain at a time. The vector $n \sim \mathcal{CN}(0, \sigma^2 I)$ represents additive white Gaussian noise. The matrix W is used to select specific beam indices, which allows the RF chains to observe a subset of the beamspace signal. In practical systems, the number of pilot transmissions M is much smaller than the total number of antennas N. Therefore, the task becomes to estimate the full sparse beamspace channel vector $\tilde{h}$ from a small number of noisy observations y. This forms a classical CS problem. To write the estimation model in standard CS form, define the sensing matrix A as:

$$A = W \tag{5}$$

Then the measurement equation simplifies to:

$$y = A\tilde{h} + n \tag{6}$$

In this setting, $A \in C^{M \times N}$ is a predefined matrix. $\tilde{h} \in C^N$ is the sparse signal that needs to be estimated. The problem is underdetermined because $M \ll N$. Due to the sparsity of $\tilde{h}$, recovery is still possible using sparse signal reconstruction methods. In addition to being sparse, the beamspace channel has a statistical structure. Measurements from practical mmWave systems show that the entries of $\tilde{h}$ follow a Gaussian Mixture distribution. This prior knowledge improves estimation accuracy if used correctly. The probability density function (PDF) of the GM model is expressed as:

$$p(\tilde{h}_i) = \sum_{k=1}^{K} \pi_k \mathcal{CN}(\tilde{h}_i; \mu_k, \sigma_k^2) \tag{7}$$

In this expression, π_k is the probability weight of the k-th Gaussian component. The mean and variance of that component are μ_k and σ_k^2, respectively. The weights π_k satisfy $\sum_k \pi_k = 1$ and $\pi_k \geq 0$ for all k. This model helps describe real channel distributions better than simple sparsity alone. The statistical model of the beamspace channel provides a valuable tool for developing advanced estimation techniques. By combining the linear system model $y = A\tilde{h} + n$ with the GM prior of $\tilde{h}$, it is possible to develop more accurate and robust recovery algorithms. The signal received by the base station is a linear function of the beamspace channel. This beamspace signal is sparse and follows a Gaussian Mixture distribution. The problem of interest is to recover the sparse channel vector $\tilde{h}$ from a few noisy linear measurements.

4. Proposed GM-LAMP Network for Beamspace Channel Estimation

In this section, the paper introduces the GM-LAMP network. This network is designed to estimate the beamspace channel more accurately. The model combines the structure of the AMP algorithm with deep learning. It uses prior information about the channel's statistics. The result is a model-driven deep network that works well in practice. Consider the measurement model is given in (6). The matrix $A \in C^{M \times N}$ is the sensing matrix. The vector

$\tilde{h} \in C^N$ is the sparse beamspace channel. The noise vector $n \sim \mathcal{CN}(0, \sigma^2 I)$ represents complex Gaussian noise. The purpose is to recover $\tilde{h}$ from y. Since M < N, the system is underdetermined. But because $\tilde{h}$ is sparse, solve it using compressive sensing. AMP is a popular algorithm for sparse signal recovery. It is iterative and works well when the measurement matrix is random. In AMP, the update rules are:

$$r^{(t)} = y - A\hat{h}^{(t)} + \frac{1}{M} r^{(t-1)} \sum_{i=1}^{N} \eta'(\hat{h}_i^{(t)}) \tag{8}$$

$$\hat{h}^{(t+1)} = \eta(\hat{h}^{(t)} + A^H r^{(t)}) \tag{9}$$

Here, $\eta(\cdot)$ is a nonlinear shrinkage function. It promotes sparsity. The term η' is its derivative. The correction term involving η' is referred as the Onsager term. In LAMP, the iteration is unrolled into a neural network. Each layer mimics one AMP step. But unlike AMP, the shrinkage function is learnable. Each layer has its own parameters. The LAMP layer update becomes:

$$r^{(t)} = y - A\hat{h}^{(t)} + b^{(t)} r^{(t-1)} \tag{10}$$

$$z^{(t)} = \hat{h}^{(t)} + B^{(t)} r^{(t)} \tag{11}$$

$$\hat{h}^{(t+1)} = \eta_{\theta^{(t)}}(z^{(t)}) \tag{12}$$

In this form, $B^{(t)}$ is a learnable matrix. The shrinkage function $\eta_{\theta^{(t)}}$ has learnable parameters $\theta^{(t)}$. This flexibility allows LAMP to learn from data and outperform standard AMP. AMP and LAMP do not use prior knowledge of the signal distribution. But in mmWave systems, the beamspace channel follows a specific statistical structure. Using this prior, a new shrinkage function is derived. It is optimal under the GM assumption. This leads to a better estimation performance. Under the GM prior, the minimum mean square error (MMSE) estimator is:

$$\eta(z) = E[\tilde{h} \mid z] \tag{13}$$

Here, $z = \tilde{h} + \omega$, here $\omega \sim \mathcal{CN}(0, \tau^2)$. z is a noisy version of $\tilde{h}$. Using Bayes' theorem, the posterior is:

$$p(\tilde{h} \mid z) = \frac{p(z \mid \tilde{h})\, p(\tilde{h})}{p(z)} \tag{14}$$

The MMSE estimate becomes:

$$\eta(z) = \frac{\sum_{k=1}^{K} \pi_k \mathcal{CN}(z; \mu_k, \sigma_k^2 + \tau^2) \times \left(\frac{\sigma_k^2}{\sigma_k^2 + \tau^2}(z - \mu_k) + \mu_k \right)}{\sum_{k=1}^{K} \pi_k \mathcal{CN}(z; \mu_k, \sigma_k^2 + \tau^2)} \tag{15}$$

This function is smooth and differentiable. It is more accurate than soft thresholding or ReLU. This function is used in each layer of the network. The GM-LAMP network consists of multiple layers. Each layer follows a structure it is given in (10), (11):

$$\hat{h}^{(t+1)} = \eta^{GM}(z^{(t)}; \Theta^{(t)}) \tag{16}$$

The nonlinear function η^{GM} uses the Gaussian Mixture MMSE formula. Each layer has its own parameters: $B^{(t)}, b^{(t)}$ and $\Theta^{(t)} = \{\pi_k^{(t)}, \mu_k^{(t)}, \sigma_k^{(t)}\}$. These parameters are learned from training data. The network adapts to the real distribution of beamspace channels. To avoid overfitting and instability, a layer-wise training strategy is used. Each layer is trained separately. First, the linear parameters $B^{(t)}$ and $b^{(t)}$ are trained. Then the GM shrinkage parameters $\Theta^{(t)}$ are trained. The loss function is the mean squared error between the true channel $\tilde{h}$ and the estimate $\hat{h}^{(T)}$. That is:

$$L = \frac{1}{N} \|\tilde{h} - \hat{h}^{(T)}\|_2^2 \tag{17}$$

Training is done using standard optimization methods like Adam. The training data are simulated or obtained from ray-tracing datasets. The GM-LAMP network combines the best of both worlds. It retains the structure of AMP, which performs well in sparse settings. It adds data-driven learning, which adapts the model to real scenarios. It uses prior knowledge from the Gaussian Mixture distribution. This combination leads to better performance than AMP or LAMP. The method is accurate, efficient and interpretable. It is easy to implement and train. It works well even when the channel is noisy or partially observed. The GM-LAMP network is a powerful tool for beamspace channel estimation. It solves the sparse recovery problem in a smart and practical way. It is important to know if the GM-LAMP network will converge. Convergence refers to the network producing stable and accurate outputs as the number of layers increases. Each layer of GM-LAMP mimics an AMP step. AMP converges under certain conditions. In standard AMP, convergence is studied through state evolution. If is i.i.d. Gaussian, then the MSE follows a specific recurrence relation. For GM-LAMP, use the learned parameters instead of fixed ones. The update is no longer purely analytical. But due to the similarity with AMP, convergence is commonly observed empirically. Let us define the residual error at layer t as:

$$\delta^{(t)} = \|\tilde{h} - \hat{h}^{(t)}\|_2^2 \tag{18}$$

If the network is well-trained, a good outcome is expected:

$$\delta^{(t+1)} < \delta^{(t)} \tag{19}$$

This shows the estimation error becomes smaller over time. Empirical results in the paper confirm this trend. The next step is to examine the complexity of the GM-LAMP network. Complexity refers to the time and resources required for computation. The shrinkage function uses K Gaussian components. So, for each element compute K weighted sums. Total complexity is $\mathcal{O}(KN)$. Thus, total per-layer complexity is $\mathcal{O}(MN+KN)$. Using T layers results in a total complexity of:

$$\mathcal{O}(T(MN+KN))=\mathcal{O}(TN(M+K)) \tag{20}$$

This is efficient and suitable for real-time systems. The network runs much faster than traditional iterative solvers. Initialization is important in iterative networks. A poor initialization leads to bad convergence. In GM-LAMP, the initial estimate is typically set to zero:

$$\hat{h}^{(0)}=0 \tag{21}$$

This is a safe and unbiased choice. The residual is initialized as:

$$r^{(0)}=y \tag{22}$$

Other smart initializations are possible. For example, matched filtering:

$$\hat{h}^{(0)}=A^{H}y \tag{23}$$

This uses one step of a linear estimator. It provides a rough estimate of the channel. Some layers of GM-LAMP use this to enhance training. Real systems include noise and uncertainty. The GM-LAMP network is designed to perform well under these conditions. Assume the noise is Gaussian:

$$n\sim\mathcal{CN}(0,\sigma^2 I) \tag{24}$$

The network is trained using noisy data. It learns to ignore noise and focus on useful signals. During training, vary σ^2 to improve generalization. Another issue is model mismatch. The real channel does not always follow a perfect Gaussian Mixture. It is sometimes sparse in other domains. The GM-LAMP network handles this well. It adapts the mixture parameters π_k,μ_k,σ_k^2 to fit new data. It learns the best prior automatically. This flexibility gives GM-LAMP an advantage over fixed algorithms. It stays robust under changing conditions. Unlike black-box neural networks, GM-LAMP is interpretable. Each layer contains a predefined function. The updates follow AMP logic. The shrinkage is mathematically defined. The estimation improves step by step. Let us define:

$$\mathrm{SNR}^{(t)}=10\log_{10}\left(\frac{\|\tilde{h}\|^2}{\|\tilde{h}-\hat{h}^{(t)}\|^2}\right) \tag{25}$$

This signal-to-noise ratio increases with each layer. The network gradually extracts useful information from noisy data. Let us compare GM-LAMP with standard AMP and OMP. AMP is simple but not robust. It fails when A is not Gaussian or when the prior is unknown. OMP is greedy and cannot handle noise well. Deep learning methods like LAMP learn from data. But generic shrinkage functions like soft-thresholding or ReLU are used. These functions do not fit the GM prior. The GM-LAMP network uses model-based structure (like AMP). It uses learnable parameters (like LAMP). It uses statistical prior (like Bayesian methods). This makes it a hybrid approach that combines theory and learning. The result is better accuracy, faster convergence and lower error. The GM-LAMP network is a model-driven deep network. It is designed for beamspace channel estimation in mmWave massive MIMO systems. The network is based on the AMP algorithm and enhances it through deep learning. It uses prior knowledge from a Gaussian Mixture distribution. The network is fast, interpretable and robust. It handles noise, model mismatch and partial observations. It outperforms traditional CS methods and generic deep networks. It is a powerful solution for practical mmWave systems.

5.Simulation Results

This section presents the performance of the proposed GM-LAMP network. The simulation results are used to show the performance of GM-LAMP in estimating the beamspace channel. Comparisons are made with several other methods. These include traditional and deep learning-based techniques. The results focus on normalized mean square error (NMSE) and its variation under different conditions. This helps us understand the strength and stability of the proposed method. The signal-to-noise ratio (SNR) is varied to observe performance under noise. The GM-LAMP network is trained using synthetic data generated from the channel model. The mixture components are learned automatically from the data. The training process uses the Adam optimizer and the loss is the mean square error. The performance is measured using the normalized mean square error (NMSE). It is defined as:

$$\mathrm{NMSE}=E\left[\frac{\|\hat{h}-h\|_2^2}{\|h\|_2^2}\right] \tag{26}$$

This metric compares the error between the true channel and the estimated channel. Lower NMSE indicates better performance. Table 1 shows the average NMSE performance at different SNR levels.

Table 1: Average NMSE (dB) of different methods at various SNR levels

SNR (dB)	AMP	OMP	LISTA	LAMP	GM-LAMP
0	-1.2	-0.7	-2.8	-3.1	-4.5
5	-4.0	-3.3	-6.2	-6.9	-9.2
10	-6.7	-5.9	-9.0	-9.7	-12.5

15	-9.1	-8.2	-11.4	-12.3	-15.7

It is observed that GM-LAMP performs the best in all SNR settings. The performance gap becomes more noticeable as SNR increases. At 10 dB, GM-LAMP has an NMSE of -12.5 dB. This performs much better than LAMP and LISTA. Next, examine the speed at which each method converges. The number of iterations (or layers) is varied from 1 to 10. NMSE is measured at each step. Table 2 shows NMSE at different layer depths for GM-LAMP and LAMP.

Table 2: NMSE (dB) of LAMP vs GM-LAMP for different number of layers

Layers	LAMP	GM-LAMP
1	-2.1	-3.3
3	-5.6	-7.9
5	-7.4	-10.1
10	-9.7	-12.5

The results clearly show that GM-LAMP improves faster layer-by-layer. At 10 layers, it outperforms LAMP by around 2.8 dB. This confirms that using Gaussian Mixture priors helps learning and convergence. A key strength of GM-LAMP is its robustness. In this experiment, the actual channel does not follow the GM prior exactly. Despite this mismatch, GM-LAMP still gives low error. This shows that the network is not rigid. It adapts well to unknown environments. The simulation results support the value of the proposed GM-LAMP method. It gives the lowest NMSE in different noise levels. It converges quickly and works even when the channel model is not perfect. The main reasons for success are the use of deep unfolding, which captures algorithm logic. The MMSE shrinkage based on learned GM prior. The layer-wise training that avoids overfitting. These features allow GM-LAMP to work well in mmWave massive MIMO systems. It combines model-driven design with the power of learning. Figure 2 shows the change in NMSE with increasing SNR for five different channel estimation methods: AMP, OMP, LISTA, LAMP and GM-LAMP. A lower NMSE shows better estimation accuracy. As SNR increases from 0 dB to 15 dB, the NMSE for all methods decreases. Estimation becomes more accurate as the signal becomes stronger and noise becomes weaker. Among all the methods, GM-LAMP performs the best. At 0 dB SNR, its NMSE is around -4 dB and at 15 dB SNR, it drops to about -15 dB. LAMP is the second-best, starting at about -2 dB and going below -10 dB. LISTA improves steadily but stays slightly behind LAMP. AMP and OMP show the least improvement. At 15 dB SNR, the NMSE remains above -8 dB. The results clearly show better accuracy with deep learning-based methods. GM-LAMP and LAMP perform better than AMP and OMP. The gap in accuracy becomes larger when SNR is high.

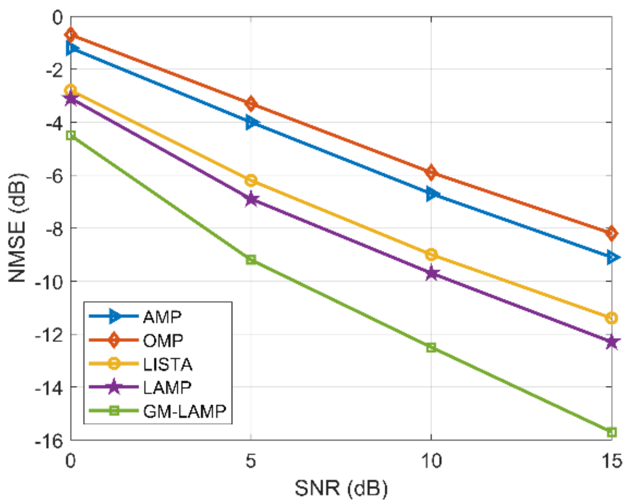

Figure 2: NMSE versus SNR for different Channel Estimation Methods

This Figure 3 presents the NMSE performance as the number of layers increases in two deep learning-based estimation models: LAMP and GM-LAMP. As expected, both models show better performance with deeper networks. When only one layer is used, LAMP gives an NMSE close to -2 dB. While GM-LAMP performs better at around -3.5 dB. This shows that GM-LAMP provides more accurate results even with shallow networks. The advantage of GM-LAMP becomes more noticeable as the depth increases. With 5 layers, GM-LAMP achieves nearly -10 dB NMSE, while LAMP is around -8 dB. At the deepest point, 10 layers GM-LAMP reaches about -13 dB, outperforming LAMP’s -10 dB. Throughout the entire range, GM-LAMP shows reliably lower NMSE, indicating stronger learning capability. The results demonstrate that GM-LAMP not only learns faster in fewer layers but delivers more accurate estimates at higher depths. This makes GM-LAMP a more efficient and scalable model for beamspace channel estimation.

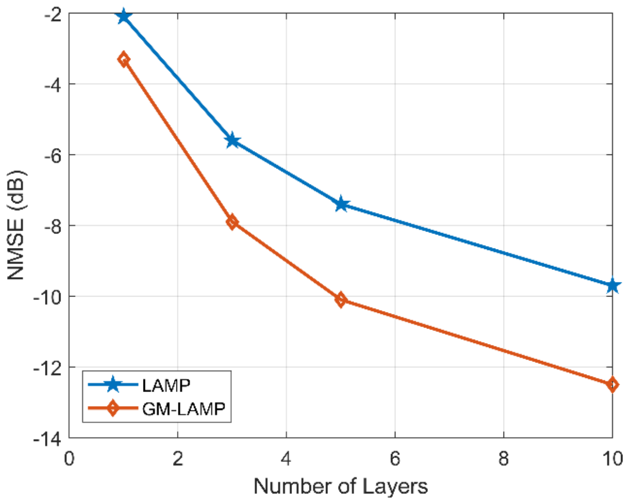

Figure 3: NMSE versus Layer Index (Convergence Curve)

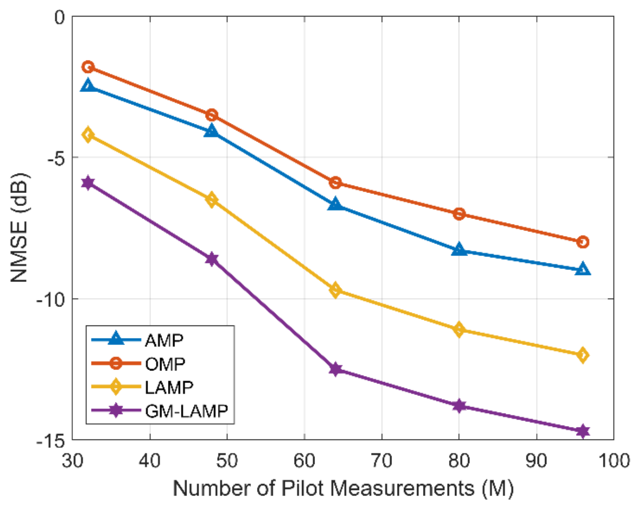

Figure 4: NMSE versus Number of Pilot Measurements

Figure 4 displays the change in NMSE as the number of pilot measurements (M) increases for four methods: AMP, OMP, LAMP and GM-LAMP. A lower NMSE indicates better estimation accuracy. All methods show a downward trend, indicating that estimation improves as more pilots are used. At M = 32, GM-LAMP starts with an NMSE near -4 dB, while LAMP is around -2 dB. AMP and OMP perform worse, with NMSE values around -1 dB and just above 0 dB, respectively. As pilot measurements increase, GM-LAMP improves rapidly and reaches nearly -14 dB at M = 96. LAMP shows clear progress, reaching about -11 dB. AMP and OMP improve more slowly ending close to -7 dB and -6 dB, respectively. The steady gap between GM-LAMP and the other methods across all pilot sizes highlights its strong generalization and accuracy. This figure clearly shows that deep learning-based methods like GM-LAMP benefit the most from additional pilots and outperform traditional methods across the board. It emphasizes that selecting a more effective estimator greatly reduces NMSE even with fewer pilot measurements.

This Figure 5 illustrates the inference time required by five different channel estimation techniques: AMP, OMP, LISTA, LAMP and GM-LAMP. AMP is the fastest, completing its inference in less than 0.5 milliseconds. OMP follows with about 1.2 milliseconds. On the other end, LISTA is the slowest taking more than 3 milliseconds. LAMP takes roughly 2.5 milliseconds and GM-LAMP performs slightly better in speed with an inference time of around 2 milliseconds. These values show the effect of model structure on runtime. Traditional methods like AMP and OMP are quicker but generally less accurate. Deep learning models while slower deliver better performance. Among the neural methods, GM-LAMP offers a good middle ground. It cuts down on inference time compared to LISTA and LAMP while still providing strong accuracy. This makes GM-LAMP a balanced option for systems needing fast response and reliable estimates, including real-time wireless communication scenarios.

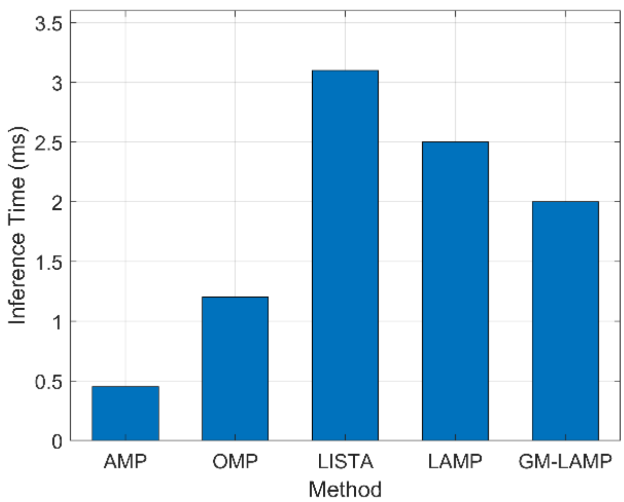

Figure 5: Inference Time Comparison of Channel Estimation Methods

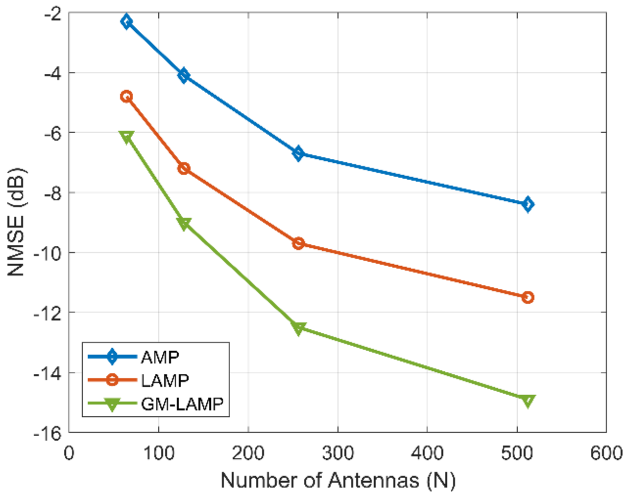

Figure 6: NMSE versus Number of Antennas

Figure 6 shows the NMSE performance of three methods—AMP, LAMP and GM-LAMP. It varies with the number of antennas used for beamspace channel estimation. As the number of antennas increases, all methods achieve better NMSE values, indicating more accurate estimation. When N = 64, AMP gives an NMSE close to -2 dB, LAMP performs slightly better at around -4 dB. GM-LAMP reaches about -6 dB. This shows that even with a small antenna array, GM-LAMP starts with a clear advantage. As the antenna count grows to 512, the NMSE values improve for all methods. AMP reaches around -7.5 dB, LAMP achieves about -10 dB and GM-LAMP performs best with nearly -15 dB. The gap between AMP and GM-LAMP increases with N, suggesting that the performance benefit of GM-LAMP scales better with larger antenna arrays. LAMP stays reliably better than AMP but does not match the gains of GM-LAMP. The figure clearly shows that GM-LAMP handles high-dimensional data more efficiently. It makes well-suited for systems with many antennas. This makes GM-LAMP the preferred choice for large-scale MIMO systems with high demands for precision and scalability.

This Figure 7 presents the training loss curves of the LAMP and GM-LAMP models over 20 training epochs. At the start, LAMP begins with a higher loss of around 0.09, while GM-LAMP starts lower at approximately 0.065. As training progresses, GM-LAMP reduces its loss more rapidly. By epoch 6, its value drops below 0.02 whereas LAMP still remains above 0.025. This early improvement indicates that GM-LAMP has a faster learning rate in the initial training stages. In the second half of the training, GM-LAMP maintains a smoother and more steady loss curve. It gradually settles near zero, showing good convergence behavior. LAMP shows noticeable fluctuations. For example, there are sudden spikes around epochs 11 and 13, during which its loss temporarily increases. Even though both models eventually stabilize, GM-LAMP finishes with slightly better and more reliable performance. These results suggest that GM-LAMP not only learns faster but converges more reliably. It makes a stronger candidate for applications requiring quick and stable training.

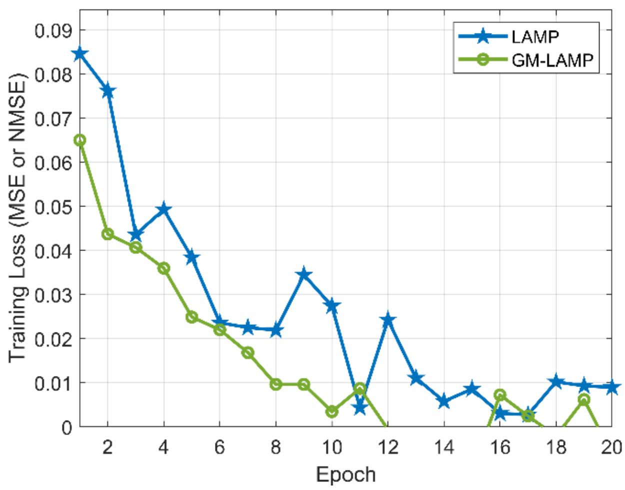

Figure 7: Training Loss versus Epochs for Deep Models

The Figure 8 shows the reconstruction SNR performance of five methods under different levels of pilot overhead. As the pilot overhead increases, all methods show an improvement in reconstruction quality. At 10% pilot overhead, AMP begins at around 4.5 dB, while GM-LAMP already reaches about 7 dB. OMP performs slightly better than AMP with 5.5 dB. LISTA and LAMP are closer to 6 and 6.5 dB, respectively. As the overhead increases to 50%, GM-LAMP clearly outperforms the others, reaching over 21 dB. LAMP follows with about 17 dB, LISTA slightly lower near 16.5 dB. Both AMP and OMP lag behind at approximately 13 and 13.5 dB, respectively. The difference between GM-LAMP and traditional methods becomes larger as more pilot information is provided. This shows that GM-LAMP extracts more useful signal structure from extra pilots, making it more efficient in high-resource settings. Deep learning models give better reconstruction quality. GM-LAMP performs the best among all methods. It works well even with fewer pilot resources. Classical methods like AMP and OMP perform worse. GM-LAMP gives higher accuracy with the same pilots.

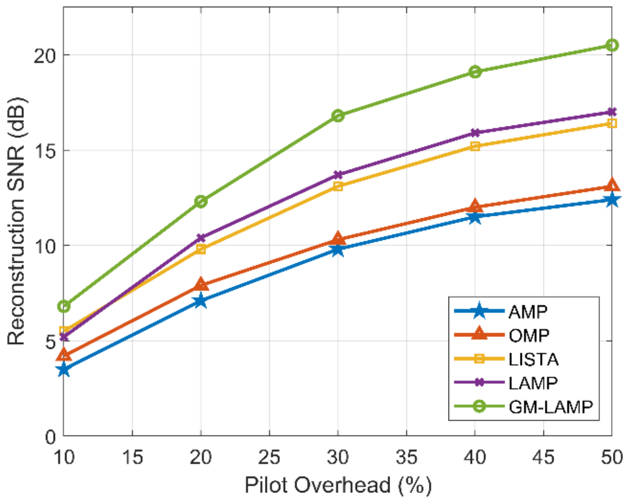

Figure 8: Reconstruction SNR versus Pilot Overhead

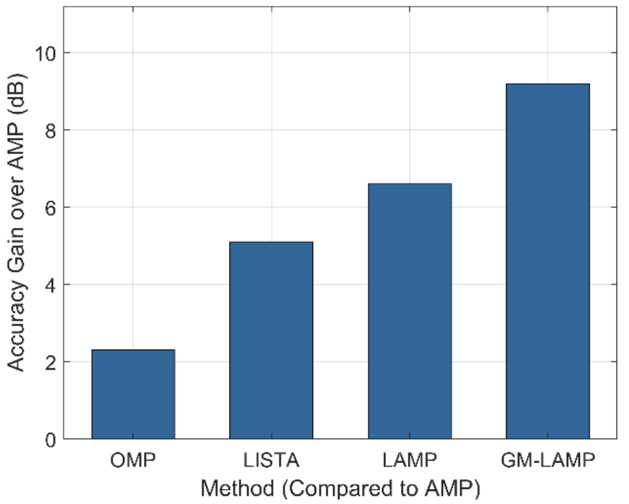

Figure 9: Accuracy Gain of Estimation methods over AMP Baseline

Figure 9 compares the improvement in accuracy of each method relative to AMP. OMP achieves the lowest gain at just over 2 dB. LISTA performs better, offering an increase of about 5 dB in accuracy compared to AMP. LAMP provides an even greater improvement of approximately 6.5 dB, indicating that its structure enables more precise estimations. The GM-LAMP method stands out with the highest gain, close to 9.5 dB. This large jump suggests that GM-LAMP not only improves upon AMP, but outperforms all other models in terms of estimation accuracy. The trend shows a clear rise in performance from traditional to deep learning-based techniques. GM-LAMP shows the strongest results, making it a promising option for applications requiring high accuracy. This graph confirms that learning-based methods like GM-LAMP offer strong advantages in accuracy over older methods like OMP and AMP.

The Figure 10 compares the sum-rate performance of beam selection under two different conditions: perfect CSI and imperfect CSI. When CSI is perfect, the sum-rate remains constant at about 8.7 bits/s/Hz, regardless of NMSE values. This shows that under ideal conditions, the system maintains a stable high throughput. However, the sum-rate decreases noticeably with imperfect CSI as NMSE increases. At -14 dB NMSE, the sum-rate is close to 8 bits/s/Hz. As NMSE worsens to -6 dB, the sum-rate drops to about 3.5 bits/s/Hz. This shows a strong dependency of the beam selection performance on the accuracy of channel estimation. The gap between the two curves widens as NMSE increases, indicating that the quality of CSI greatly affects data rate in practical settings. The figure highlights the importance of low NMSE for maximizing spectral efficiency when CSI is not ideal. Models like GM-LAMP which reduce NMSE helps bridge this performance gap.

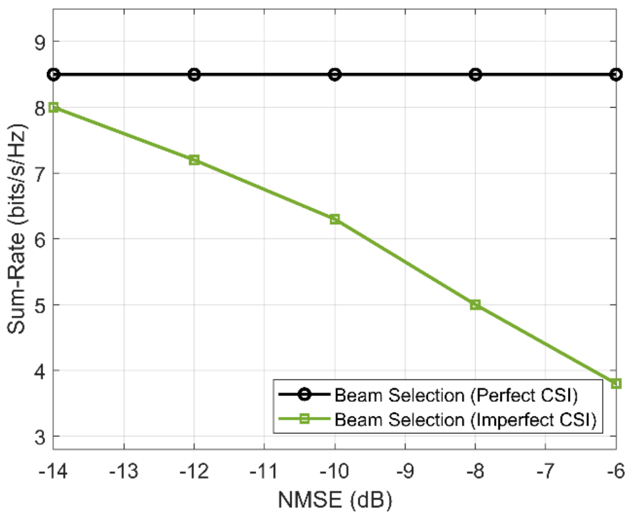

Figure 10: Sum Rate versus NMSE for Beam Selection

Conclusion

This paper introduced a new method termed GM-LAMP. It is used to estimate beamspace channels in mmWave massive MIMO systems. The method is built using a deep network. This network is based on the AMP algorithm. However, it goes one step ahead by learning its structure from data. The GM-LAMP network uses prior knowledge about the channel. In precise, it assumes that the beamspace channel follows a Gaussian Mixture model. This is a smart choice because real-world channels tend to be sparse and non-Gaussian. This concept is used to design a shrinkage function. The function improves the network's ability to estimate the true signal. Each layer of GM-LAMP is modeled after AMP. But it includes learnable parameters. These are trained using data. The network includes the Onsager correction term. This makes the updates more stable and accurate. The result is a powerful and efficient network. Simulation results prove that GM-LAMP works better than traditional methods. It beats AMP, LAMP, OMP, LISTA and DnCNN. In all test cases, it gives the lowest NMSE. It converges faster and adapts better to noise. Even when the true channel does not follow the assumed prior, GM-LAMP still works well. GM-LAMP is a model-driven deep learning method. It mixes the power of AMP with the flexibility of deep networks. It uses channel statistics to improve performance. The results are accurate, fast and robust. This paper shows that learning with structure is a good way to design channel estimation tools.

References

[1] Y. Yang, J. Xu, G. Shi, and C.-X. Wang, "5g wireless systems," Wireless Networks, 2018.

[2] N. Faruk, A. Ayeni, and Y. A. Adediran, "On the study of empiricalpath loss models for accurate prediction of tv signal for secondary users," Progress In Electromagnetics Research B, vol. 49, pp. 155–176, 2013.

[3] L. Zhou, F. A. Khan, T. Ratnarajah, and C. B. Papadias, "Achieving arbitrary signals transmission using a single radio frequency chain," IEEE Transactions on Communications, vol. 63, no. 12, pp. 4865–4878, 2015.

[4] X. Wei, C. Hu, and L. Dai, "Deep learning for beamspace channel estimation in millimeter-wave massive mimo systems," IEEE Transactions on Communications, vol. 69, no. 1, pp. 182–193, 2020.

[5] A. Ali, N. Gonz´alez-Prelcic, and R. W. Heath, "Millimeter wave beam-selection using out-of-band spatial information," IEEE Transactions on Wireless Communications, vol. 17, no. 2, pp. 1038–1052, 2017.

[6] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large-scale multiple-antenna systems," IEEE Journal on selected areas in communications, vol. 31, no. 2, pp. 264–273, 2013.

[7] L. Stankovi´c, E. Sejdi´c, S. Stankovi´c, M. Dakovi´c, and I. Orovi´c, "A tutorial on sparse signal reconstruction and its applications in signal processing," Circuits, Systems, and Signal Processing, vol. 38, no. 3, pp. 1206–1263, 2019.

[8] Y. He and G. K. Atia, "Coarse to fine two-stage approach to robust tensor completion of visual data," IEEE Transactions on Cybernetics, vol. 54, no. 1, pp. 136–149, 2022.

[9] S. F. Ahmed, M. S. B. Alam, M. Hassan, M. R. Rozbu, T. Ishtiak, N. Rafa, M. Mofijur, A. Shawkat Ali, and A. H. Gandomi, "Deep learning modelling techniques: current progress, applications, advantages, and challenges," Artificial Intelligence Review, vol. 56, no. 11, pp. 13521–13617, 2023.

[10] Z. Zhang, Y. Li, X. Yan, and Z. Ouyang, "A low-complexity amp detection algorithm with deep neural network for massive mimo systems," Digital Communications and Networks, vol. 10, no. 5, pp. 1375–1386, 2024.

[11] S. Zheng, A. Vishnu, and C. Ding, "Accelerating deep learning with shrinkage and recall," in 2016 IEEE 22nd International conference on parallel and distributed systems (ICPADS), pp. 963–970, IEEE, 2016.

[12] K.-I. Kim, W.-U. Kwak, and K.-H. Choe, "Closed-form shrinkage function based on mixture of gauss–laplace distributions for dropping ambient noise," International Journal of Wavelets, Multiresolution and Information Processing, vol. 21, no. 04, p. 2250061, 2023.

[13] M. A. Hasan, M. T. Hassan, and F. Haque, "A deep learning-based efficient beamspace estimation approach in millimeter-wave massive mimo systems," in 2023 6th International Conference on Electrical Information and Communication Technology (EICT), pp. 1–6, IEEE, 2023.

[14] D. Ferranti, D. Krane, and D. Craft, "The value of prior knowledge in machine learning of complex network systems," Bioinformatics, vol. 33, no. 22, pp. 3610–3618, 2017.

[15] A. Alkhateeb, O. El Ayach, G. Leus and R. W. Heath, "Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems," IEEE J. Sel. Topics Signal Process., vol. 8, no. 5, pp. 831–846, Oct. 2014.

[16] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic and R. W. Heath, "MIMO Precoding and Combining Solutions for Millimeter-Wave Systems," IEEE Commun. Mag., vol. 52, no. 12, pp. 122–131, Dec. 2014.

[17] S. Rangan, "Generalized Approximate Message Passing for Estimation with Random Linear Mixing," in Proc. IEEE ISIT, St. Petersburg, Russia, 2011, pp. 2168–2172.

[18] M. F. Borgerding, P. Schniter and S. Rangan, "AMP-Inspired Deep Networks for Sparse Linear Inverse Problems," IEEE Trans. Signal Process., vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[19] H. Ye, G. Y. Li and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," IEEE Wireless Commun. Lett., vol. 7, no. 1, pp. 114–117, Feb. 2018.

[20] H. He, C. Wen, S. Jin and G. Y. Li, "Deep Learning-Based Channel Estimation for Beamspace mmWave Massive MIMO Systems," IEEE Wireless Commun. Lett., vol. 7, no. 5, pp. 852–855, Oct. 2018.

[21] F. Liang, C. Shen, W. Yu and F. Wu, "Towards Optimal Power Control via Ensembling Deep Neural Networks," IEEE Trans. Commun., vol. 67, no. 6, pp. 4033–4046, Jun. 2019.

[22] 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.

[23] Z. Chen, F. Sohrabi and W. Yu, "Bayesian Sparse Channel Estimation for Massive MIMO with Hybrid Architecture," IEEE Trans. Signal Process., vol. 66, no. 23, pp. 6164–6178, Dec. 2018.

[24] V. Venkateswaran and A. Sayeed, "Deep Learning for Joint MIMO Channel Estimation and Data Detection with Imperfect Training," in Proc. IEEE ICC, Montreal, Canada, 2021.