Conditional Diffusion Model-Driven Massive MIMO Iterative Detection

Keke Ying*, Zhen Gao†*, De Mi‡, Ziwei Wan§, Sheng Chen¶, Tony Q.S. Quek‖, and H. Vincent Poor**
*School of Information and Electronics, Beijing Institute of Technology, Beijing, China
† State Key Laboratory of CNS/ATM, Beijing Institute of Technology, Beijing, China
‡ College of Computing, Birmingham City University, Birmingham, United Kingdom
§ Yangtze Delta Region Academy, Beijing Institute ofTechnology (Jiaxing), Jiaxing, China
¶ School of Computer Science and Technology, Ocean University of China, Qingdao, China
‖ Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore
** Department of Electrical and Computer Engineering, Princeton University, Princeton, USA
(Email: ykk@bit.edu.cn, gaozhen16@bit.edu.cn)

***Abstract*—To ensure future high-quality and reliable massive communication during 6G uplink transmissions, we propose a conditional diffusion model-driven massive MIMO detector, which can iteratively estimate channels and detect data for the uplink multiuser access. This approach utilizes a generative diffusion model to learn the score function of the joint posterior by integrating the prior distribution with the likelihood derived from the transmission model. The prior distribution is obtained either by learning from channel statistics or through analytical derivation from the symbol constellation. By employing noise matching initialization and an asynchronous annealed Langevin dynamics (ALD) sampling scheme, the receiver alternates efficiently between score-based channel estimation and data detection, thus avoiding traps of local minima. Simulation results demonstrate that this iterative diffusion process outperforms Bayesian-based and existing synchronous ALD channel estimation and data detection schemes in multiuser uplink scenarios.**

***Index Terms*—Deep learning, diffusion model, iterative receiver, massive MIMO, multiuser detection.**

I. Introduction

Establishing high-quality and reliable connections is a crucial goal for future 6G massive communication scenarios. Specifically, multiuser uplink access [1] is a critical step in establishing initial connections between user terminals (UTs) and the base station (BS), particularly in 6G massive communication where multiple short data packets uploaded by Internet of Things (IoT) devices are superimposed at the receiver. To reduce access delay for multiple users, the user side typically employs a typical transmission format with short pilot signals combined with data payloads. Meanwhile, the BS often utilizes a large array of antennas to conduct multi-user detection over the same time-frequency resources. To ensure reliable detection and quality of service (QoS), high-performance channel estimation (CE) and data detection (DD) at the BS are fundamental problems.

CE and DD can be approached as linear detection or estimation problems. Established algorithms for these tasks include least squares (LS) for CE and zero-forcing (ZF) for DD. When the observation dimension, such as pilot length or the number of antennas, is limited, leveraging prior structured characteristics of the channel, like sparsity, and the known constellation distribution of the data is crucial for enhancing performance. Consequently, algorithms such as compressed sensing [2] and Bayesian message passing [3] are effective methods to improve CE and DD performance.

To further accelerate iterative processes based on compressed sensing or message passing, methods based on deep unfolding [4]–[7] and purely data-driven approaches [8] are extensively employed. These methods offer the advantage of having a low number of neural parameters and model-driven interpretability. However, they face inherent limitations: the dimensions of network parameters are tied to those of the system configuration, which often necessitates network retraining when pilot lengths or the number of users vary, thus lacking scalability.

In recent years, diffusion model-based approaches [9] have been widely adopted for inverse problem solving. Unlike Bayesian algorithms that depend on assumed priors, generative models [10], [11] learn complex signal distributions in a data-driven manner. They require only self-supervised pre-training on relevant signal dimensions, without full transmission–reception datasets, improving data efficiency and adaptability across observation scenarios. Specifically, in diffusion models, the score matching with Langevin dynamics (SMLD) is a notable framework [10] that estimates the *score function* (defined as the gradient of the signal's log-probability) using either analytical or learned methods. Using this score, annealed Langevin dynamics (ALD) gradually samples signals from a high-dimensional distribution with a sequence of decreasing noise perturbations.

Conditional diffusion models, using SMLD frameworks have been applied to tackle linear inverse problems in channel estimation [12] and data detection [13], [14]. Furthermore, these models have been extended to joint channel estimation and data detection [15], as well as quantized channel estimation [16]. They exhibit superior performance compared to traditional algorithms and contemporary deep learning techniques [4], [5], [8].

In the multiuser uplink transmission scheme with short pilots and data payloads, the diffusion-based joint CE and DD (JCEDD) method [15] notably enhances detection and resource efficiency but suffers from local optima and high iteration costs.

To address this, we improve JCEDD initialization and iteration, extending it to multi-user uplink access. The contributions of this paper are listed as follows:

- We reformulate JCEDD as the reverse diffusion sampling process. By employing pilot signals for channel initialization and diffusion noise matching, we iteratively refine channels and data symbols based on the estimated initial point, effectively reducing the number of diffusion steps.
- We introduce an asynchronous CE-DD iteration to prevent DD from becoming trapped in local optima, thereby achieving superior JCEDD performance compared to existing Langevin sampling-based [15] and Bayesian-type JCEDD methods [5].

Simulations conducted under complex 3GPP channel conditions confirm that the proposed conditional diffusion model-driven iterative detector achieves more reliable JCEDD performance compared to benchmark algorithms. This ensures high QoS in multi-user uplink detection with low pilot overhead.

II. Problem Formulation and Diffusion Process

Consider a narrowband uplink transmission scenario where K UEs simultaneously access a single BS. The BS is equipped with a uniform planar array (UPA) of $M = M_x \times M_y$ antennas, with M_x and M_y antennas positioned along the x-axis and y-axis, respectively. Each UE is assumed to use a single antenna. For the k-th UE, the uplink frame consists of L_p pilot symbols $\boldsymbol{p}_k \in \mathbb{C}^{L_p}$ followed by L_d data symbols $\boldsymbol{x}_k \in \mathbb{C}^{L_d}$. These pilot and data symbols form a frame $\boldsymbol{s}_k = \left[\boldsymbol{p}_k^{\mathrm{T}}, \boldsymbol{x}_k^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{L}$ of length $L = L_p + L_d$. The channel vector between the k-th UE and the BS is denoted by $\boldsymbol{h}_k \in \mathbb{C}^{M}$, which is assumed to remain constant throughout the entire transmission frame. The received signal matrix $\boldsymbol{Y} \in \mathbb{C}^{L\times M}$ over a frame from all UEs is expressed as:

$$\boldsymbol{Y} = \sum_{k=1}^{K} \boldsymbol{s}_k \boldsymbol{h}_k^{\mathrm{T}} + \boldsymbol{W} = \boldsymbol{S}\boldsymbol{H} + \boldsymbol{W}, \tag{1}$$

where we define $\boldsymbol{S} \triangleq \left[\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_K\right] \in \mathbb{C}^{L\times K}$, $\boldsymbol{H} \triangleq \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K\right]^{\mathrm{T}} \in \mathbb{C}^{K\times M}$, and $\boldsymbol{W} \in \mathbb{C}^{L\times M}$ is the noise matrix, whose elements are symmetric complex additive Gaussian white noise (AWGN) with zero mean and variance σ_n^2. By splitting the pilot and data components in $\boldsymbol{S}$ into two parts, i.e., $\boldsymbol{P} \triangleq \left[\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_K\right] \in \mathcal{X}^{L_p\times K}$ and $\boldsymbol{X} \triangleq \left[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\right] \in \mathcal{X}^{L_d\times K}$, we have $\boldsymbol{S} = \left[\boldsymbol{P}^{\mathrm{T}}, \boldsymbol{X}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathcal{X}^{L\times K}$ and $\boldsymbol{Y} = \left[\boldsymbol{Y}_p^{\mathrm{T}}, \boldsymbol{Y}_d^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{L\times M}$, where $\mathcal{X}$ denotes the quadrature amplitude modulation (QAM) constellation set, $\boldsymbol{Y}_p \in \mathbb{C}^{L_p\times M}$ and $\boldsymbol{Y}_d \in \mathbb{C}^{L_d\times M}$ correspond to the received pilot and data components, respectively. We also denote $\boldsymbol{W} = \left[\boldsymbol{W}_p^{\mathrm{T}}, \boldsymbol{W}_d^{\mathrm{T}}\right] \in \mathbb{C}^{L\times M}$ with $\boldsymbol{W}_p \in \mathbb{C}^{L_p\times M}$ and $\boldsymbol{W}_d \in \mathbb{C}^{L_d\times M}$.

Given the observations $\boldsymbol{Y}$, diffusion-based methods conceptualize DD and CE as the processes of sampling signals from the conditional posterior distributions. This can be expressed as:

$$\hat{\boldsymbol{H}} \sim p\left(\boldsymbol{H}|\boldsymbol{Y}; \hat{\boldsymbol{X}}, \boldsymbol{P}\right), \hat{\boldsymbol{X}} \sim p\left(\boldsymbol{X}|\boldsymbol{Y}; \hat{\boldsymbol{H}}, \boldsymbol{P}\right). \tag{2}$$

For any distribution $p_{\text{data}}(\mathbf{x})$ we wish to sample, general diffusion-based models can be interpreted as a continuous-time forward process that corrupts the data $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0) = p_{\text{data}}(\boldsymbol{x})$ using varying noise scales, forming a known prior distribution $p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{0}, \sigma_T^2 \boldsymbol{I})$. A backward process is then applied to transform this prior distribution back to the data distribution by progressively removing noise. Let $t \in (0, T]$ represent the continuous-time variable. These two processes are mathematically characterized by a forward stochastic differential equation (SDE) [17], $\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}$, and a corresponding *backward SDE*, $\mathrm{d}\boldsymbol{x} = \left(\boldsymbol{f}(\boldsymbol{x}, t) - g(t)^2 \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)\right)\mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}$. Here, $\boldsymbol{f}(\boldsymbol{x}, t) : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ is the predefined drift function, and $g(t) : \mathbb{R} \to \mathbb{R}$ is referred to as the diffusion function. Additionally, $\boldsymbol{w}$ is a standard Wiener process, defined by $\mathrm{d}\boldsymbol{w} = \boldsymbol{z}(t)\mathrm{d}t$, where $\boldsymbol{z}(t) \sim \mathcal{N}(\boldsymbol{z}(t); \boldsymbol{0}, \boldsymbol{I})$.

Specifically, when the drift function is zero and the diffusion function is represented by a series of scheduled noise levels, the time interval $(0, T]$ can be divided into N discrete grids. The ALD then serves as a type of first-order solver for the backward SDE process, which is given by

$$\boldsymbol{x}_{i-1} = \boldsymbol{x}_i + \epsilon_i \nabla_{\boldsymbol{x}_i} \log p(\boldsymbol{x}_i) + \sqrt{2\epsilon_i}\boldsymbol{w}, \tag{3}$$

where $\epsilon_i = 2\sigma_i \left(r\|\boldsymbol{w}\|_2 / \|\nabla_{\boldsymbol{x}_i} \log p(\boldsymbol{x}_t)\|_2\right)^2$ is the step size in the i-th iteration of the ALD algorithm [17], $\sigma_i = \sigma_t|_{t=iT/N}$ represents the diffusion noise level derived from the predefined noise level scheduling function, and $r = 0.3$ is an empirically chosen hyperparameter. Intuitively, Equ. (3) resembles a gradient descent form, aiming to guide the sample movement in the direction of $\nabla_{\boldsymbol{x}t} \log p(\boldsymbol{x}t)$ so that the final point converges to the data distribution $p(\boldsymbol{x}_0)$ with maximum probability.

Based on the diffusion process described above, both CE and DD can be reformulated as conditional sampling processes by replacing the target score function $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$ with the corresponding posterior score using the conditional distribution in Equ. (2), which will be further explained in the subsequent section.

III. Conditional Diffusion Receiver for JCEDD

In this section, we initially examine the CE problem by detailing the calculation of posterior score functions within the ALD sampler process. Subsequently, we apply the ALD sampler to the DD problem. Finally, we integrate the CE and DD methods to propose a diffusion-based asynchronous iterative algorithm for JCEDD.

A. Channel Estimation

1) Channel Prior Score Function: To apply the ALD process (3) to our CE problem, it is essential to derive the posterior score function, $\nabla_{\boldsymbol{H}} \log p\left(\boldsymbol{H}|\boldsymbol{Y}; \hat{\boldsymbol{X}}, \boldsymbol{P}\right)$. Assuming the data $\hat{\boldsymbol{X}}$ is known and irrelevant variables are omitted, the posterior score function can be derived using Bayes' theorem as follows:

$$\nabla_{\boldsymbol{H}_t} \log p\left(\boldsymbol{H}_t|\boldsymbol{Y}\right) = \nabla_{\boldsymbol{H}_t} \log p\left(\boldsymbol{Y}|\boldsymbol{H}_t\right) + \nabla_{\boldsymbol{H}_t} \log p\left(\boldsymbol{H}_t\right), \tag{4}$$

where the *likelihood score function* $\nabla_{\boldsymbol{H}_t} \log p\left(\boldsymbol{Y}|\boldsymbol{H}_t\right)$, is determined by the signal transmission model (1), and the *prior score*

function $\nabla_{\boldsymbol{H}} \log p\left(\boldsymbol{H}_t\right)$ is determined by the prior distribution of the channel dataset.

Let $\boldsymbol{h}_k \triangleq [\boldsymbol{H}]_{[k,:]}^{\mathrm{T}} \in \mathbb{C}^M$. Given that the prior distributions of each active UE's channel are assumed to be independent of one another, it follows that $\log p(\boldsymbol{H}_t) = \log \prod_{k=1}^{K} p(\boldsymbol{h}_{k,t}) = \sum_{k=1}^{K} \log p(\boldsymbol{h}_{k,t})$. Consequently, the prior score function can be computed separately for each UE, expressed as:

$$\nabla_{\boldsymbol{H}_t} \log p\left(\boldsymbol{H}_t\right) = \Big[\nabla_{\boldsymbol{h}_{1,t}} \log p\left(\boldsymbol{h}_{1,t}\right), \nabla_{\boldsymbol{h}_{2,t}} \log p\left(\boldsymbol{h}_{2,t}\right), \cdots, \nabla_{\boldsymbol{h}_{K,t}} \log p\left(\boldsymbol{h}_{K,t}\right)\Big]^{\mathrm{T}}. \tag{5}$$

Thus, when the channels of all UEs are generated from the same dataset distribution $\boldsymbol{p}_{\text{data}}(\boldsymbol{h})$, we only need to focus on the prior score function of an individual UE.

For simplicity, we omit the UE index k and define $\boldsymbol{p}(\boldsymbol{h}_0) \triangleq \boldsymbol{p}_{\text{data}}(\boldsymbol{h})$. The objective is to derive the score function $\nabla_{\boldsymbol{h}_t} \log p(\boldsymbol{h}_t)$, where $p(\boldsymbol{h}_t)$ represents the data distribution diffused by noise σ_t^2 at time t. For complex practical channel distributions, the prior score function lacks a closed-form solution. Fortunately, as demonstrated by the study [18], employing a neural network to learn the score function $\nabla_{\boldsymbol{h}_t} \log p(\boldsymbol{h}_t)$ is equivalent to denoising score matching (DSM). This can be achieved by training a time-dependent neural network $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{h}_t, t)$ using the following loss function:

$$\mathcal{L}_s = \mathbb{E}_t \left\{ \lambda_t \mathbb{E} \left(\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{h}_t, t) + \frac{\boldsymbol{h}_t - \boldsymbol{h}_0}{\sigma_t^2} \right\|_2^2 \right) \right\}, \tag{6}$$

where σ_t is the noise scheduling function at time t, which diffused the initial data point with $p\left(\boldsymbol{h}_t|\boldsymbol{h}_0\right) = \mathcal{N}\left(\boldsymbol{h}_t; \boldsymbol{h}_0, \sigma_t^2 \boldsymbol{I}\right)$, $\lambda_t \propto 1/\mathbb{E}[\|\nabla_{\boldsymbol{h}_t} \log p(\boldsymbol{h}_t|\boldsymbol{h}_0)\|_2^2]$ is a positive weight function, and $\boldsymbol{\theta}$ collects the neural network parameters. Given the powerful expressive capability of the noise conditional score network (NCSN++) [17] in image generation tasks, and considering that the channel can be viewed as a special kind of image after transforming the original M-dimensional channel vector reshaping into a real representation with $2 \times M_x \times M_y$ dimensions, NCSN++ is employed to construct $\boldsymbol{s}_{\boldsymbol{\theta}^*}(\boldsymbol{h}(t), t)$.

2) Channel Likelihood Score Function: Since channels are treated as three-dimensional real tensors in prior score learning, the transmission model (1) can also be converted into a corresponding real-valued form. For simplicity, the notation $\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{H} + \boldsymbol{W}$ is maintained. Note that in the adopted SMLD framework, $p\left(\boldsymbol{H}_t|\boldsymbol{H}_0\right) \approx \mathcal{N}\left(\boldsymbol{H}_t; \boldsymbol{H}_0, \sigma_t^2 \boldsymbol{I}\right)$. Consequently, the transmission model can be equivalently expressed as:

$$\boldsymbol{Y} = \boldsymbol{S}\left(\boldsymbol{H}_t + \sigma_t^2 \boldsymbol{Z}\right) + \boldsymbol{W}, \tag{7}$$

where $\boldsymbol{Z} \sim \mathcal{N}\left(\boldsymbol{Z}; \boldsymbol{0}, \boldsymbol{I}\right)$ represents the diffusion noise superimposed on the original channel $\boldsymbol{H}_0$. Therefore, the corresponding transition probability is given by: $p\left(\boldsymbol{Y}|\boldsymbol{H}_t\right) \sim \mathcal{N}\left(\boldsymbol{Y}; \boldsymbol{S}\boldsymbol{H}_t, \sigma_n^2 \boldsymbol{I} + \sigma_t^2 \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}\right)$. Taking the gradient of the log-likelihood, we obtain:

$$\nabla_{\boldsymbol{H}_t} \log p\left(\boldsymbol{Y}|\boldsymbol{H}_t\right) = \boldsymbol{S}^{\mathrm{T}}\left(\sigma_n^2 \boldsymbol{I} + \sigma_t^2 \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}\right)^{-1}\left(\boldsymbol{Y} - \boldsymbol{S}\boldsymbol{H}_t\right). \tag{8}$$

In the considered JCEDD problem,the matrix $\boldsymbol{S} = \left[\boldsymbol{P}^{\mathrm{T}}, \hat{\boldsymbol{X}}^{\mathrm{T}}\right]^{\mathrm{T}}$ encompasses both the pilot symbol $\boldsymbol{P}$ and the estimated data symbol $\hat{\boldsymbol{X}}$ during the iterative process. The inclusion of the estimated data symbol enhances CE. When no data symbol is used, the problem degrades to the conventional pure pilot-aided CE problem.

For the practical implementation of the algorithm, we employ singular value decomposition (SVD) to factorize the matrix $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$. This approach circumvents inverting non-diagonal matrices in Equ (8) and subsequently reduces computational complexity. The gradient of the log-likelihood with respect to $\boldsymbol{H}_t$ is then given by:

$$\nabla_{\boldsymbol{H}_t} \log p\left(\boldsymbol{Y}|\boldsymbol{H}_t\right) = \boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\left(\sigma_n^2 \boldsymbol{I} + \sigma_t^2 \boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\mathrm{T}}\right)^{-1} \tag{9}$$

$$\times \left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{H}_t\right), \tag{10}$$

By substituting Equ.(5) and Equ.(8) into Equ. (4), we derive the posterior score of the channel. This result can then be integrated into the subsequent ALD process described in Equ. (3) to facilitate posterior sampling.

B. Data Detection

Upon reviewing the formulas (1), the DD problem can be characterized as $\boldsymbol{Y}_d = \boldsymbol{X}_d\boldsymbol{H} + \boldsymbol{W}_d$. By transposing the formula and representing the model in a real-valued form, the signal transmission model can be reformulated as:

$$\bar{\boldsymbol{Y}}_d = \bar{\boldsymbol{H}}\bar{\boldsymbol{X}} + \bar{\boldsymbol{W}}_d, \tag{11}$$

where $\bar{\boldsymbol{Y}}_d$, $\bar{\boldsymbol{H}}$, $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{W}}_d$ are the transposed counterparts of the corresponding variables in real-valued form. Similar to the CE problem, assuming that the channel $\bar{\boldsymbol{H}}$ is known, data symbols can be sampled from the posterior distribution $p(\bar{\boldsymbol{X}}|\bar{\boldsymbol{Y}}_d)$ using ALD, which also involves calculating the prior score and the likelihood score of the data symbol.

1) Data Prior Score Function: Since the data symbols are uniformly sampled from a QAM constellation, the prior score $\nabla_{\bar{\boldsymbol{X}}_t} \log p\big(\bar{\boldsymbol{X}}_t\big)$ at time t-th iteration step can be calculated using Tweedie's formula [19], i.e.,

$$\nabla_{\bar{\boldsymbol{X}}_t} \log p(\bar{\boldsymbol{X}}_t) = \frac{\mathbb{E}\big\{\bar{\boldsymbol{X}}_0|\bar{\boldsymbol{X}}_t\big\} - \bar{\boldsymbol{X}}_t}{\tau_t^2}, \tag{12}$$

where τ_t is the scheduled perturbation noise function for data symbol. In (12), the posterior distribution can be derived using Bayes' rule as: $p(\bar{\boldsymbol{X}}_0|\bar{\boldsymbol{X}}_t) = \frac{p(\bar{\boldsymbol{X}}_t|\bar{\boldsymbol{X}}_0)p(\bar{\boldsymbol{X}}_0)}{\sum_{\bar{\boldsymbol{X}}_0 \sim \mathcal{X}} p(\bar{\boldsymbol{X}}_t|\bar{\boldsymbol{X}}_0)p(\bar{\boldsymbol{X}}_0)}$, where $p(\bar{\boldsymbol{X}}_0) = \frac{1}{|\mathcal{X}|_c}$. Then, we can calculate the posterior mean $\mathbb{E}\big\{\bar{\boldsymbol{X}}_0|\bar{\boldsymbol{X}}_t\big\}$ and substitute the result into the Equ. (12), which yields

$$\nabla_{\bar{\boldsymbol{X}}_t} \log p(\bar{\boldsymbol{X}}_t) = \frac{1}{\tau_t^2}\left(\frac{\sum\limits_{\bar{\boldsymbol{X}}_0 \sim \mathcal{X}} \bar{\boldsymbol{X}}_0 e^{-\frac{\|\bar{\boldsymbol{X}}_t - \bar{\boldsymbol{X}}_0\|^2}{2\tau_t^2}}}{\sum\limits_{\bar{\boldsymbol{X}}_0 \sim \mathcal{X}} e^{-\frac{\|\bar{\boldsymbol{X}}_t - \bar{\boldsymbol{X}}_0\|^2}{2\tau_t^2}}} - \bar{\boldsymbol{X}}_t\right). \tag{13}$$

2) Data Likelihood Score Function: Similar to the channel likelihood score, data likelihood score can be derived according to the transmission model (11), which is given by:

$$\nabla_{\bar{\boldsymbol{X}}_t} \log p\left(\bar{\boldsymbol{Y}}_d|\bar{\boldsymbol{X}}_t\right) = \boldsymbol{V}_h\boldsymbol{\Sigma}_h^{\mathrm{T}}\left(\sigma_n^2 \boldsymbol{I}_{2M} + \tau_t^2 \boldsymbol{\Sigma}_h\boldsymbol{\Sigma}_h^{\mathrm{T}}\right)^{-1}$$

$$\times \left(\boldsymbol{U}_h^T\bar{\boldsymbol{Y}}_d - \boldsymbol{\Sigma}_h\boldsymbol{V}_h^{\mathrm{T}}\bar{\boldsymbol{X}}_t\right), \tag{14}$$

where we have used the SVD $\bar{\boldsymbol{H}} = \boldsymbol{U}_h \boldsymbol{\Sigma}_h \boldsymbol{V}_h^{\mathrm{T}}$ to reduce the complexity during the iteration. Combining Equ. (13) and Equ.(14), we obtain the posterior score of the channel, which can be used for subsequent ALD process in Equ. (3).

C. Overall Algorithm

In the derivation of the channel and data scores, the data $\boldsymbol{X}$ is assumed to be known during CE, whereas the channel $\boldsymbol{H}$ is presumed to be known during DD. In practical transmission scenarios, however, only the pilot $\boldsymbol{P}$ and the received signals $\boldsymbol{Y}$ are available. Consequently, we propose an iterative algorithm to perform JCEDD. The proposed ALD sampler-based JCEDD method is illustrated in Fig. 1, and the algorithmic framework is outlined in Algorithm 1.

The primary issues to be addressed are as follows: 1) A conventional ALD sampling algorithm initiates the process with sampling from random noise, which can lead to a high number of iterations. 2) In the current ALD-based JCEDD approach [15], each CE iteration is followed by a DD iteration. Due to the differing distributions of the channel and data symbols, their convergence rates are inconsistent, often causing the DD to become trapped in local optima before the CE iterations have fully converged. The proposed sampling mechanism introduces several improvements to address these challenges.

- We perform linear minimum mean square error (LMMSE) estimation [5] of the channel matrix, $\boldsymbol{H}$, using the relationship $\boldsymbol{Y}_p = \boldsymbol{P}\boldsymbol{H} + \boldsymbol{W}_p$. The initial diffusion perturbation noise variance, which aligns with the estimated channel variance, serves as the starting point for ALD sampling.

Algorithm 1 Diffusion-Based Asynchronous JCEDD Algorithm

Require: $\boldsymbol{P}$, $\boldsymbol{Y}$, σ_n^2, pre-trained denoising network $\boldsymbol{s}_{\boldsymbol{\theta}^*}$, noise scheduling functions σ_t and τ_t, step numbers N_H and N_D, hyper-parameters λ_h and λ_x;

1: **Initialize:** Acquire initial CE $\hat{\boldsymbol{H}}$ via LMMSE, initial DD $\hat{\boldsymbol{X}}$ via ZF, and start point i^* via (15);
2: **Calculate:** $\boldsymbol{S} = \left[\boldsymbol{P}^{\mathrm{T}}, \hat{\boldsymbol{X}}^{\mathrm{T}}\right]^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$;
3: **for** $i = i^*$ to 1 **do**
4: Calculate $\nabla_{\boldsymbol{H}_i} \log p\left(\boldsymbol{H}_i|\boldsymbol{Y}\right) = \lambda_h \nabla_{\boldsymbol{H}_i} \log p\left(\boldsymbol{Y}|\boldsymbol{H}_i\right) + \nabla_{\boldsymbol{H}_i} \log p\left(\boldsymbol{H}_i\right)$;
5: $\boldsymbol{H}_{i-1} = \boldsymbol{H}_i + \epsilon_{i,H} \nabla_{\boldsymbol{H}_i} \log p(\boldsymbol{H}_i|\boldsymbol{Y}) + \sqrt{2\epsilon_{i,H}}\boldsymbol{W}$;
6: **if** $i \% N_{\text{update}} == 0$ **then**
7: Calculate $\boldsymbol{H}_{i-1} = \boldsymbol{U}_h \boldsymbol{\Sigma}_h \boldsymbol{V}_h^{\mathrm{T}}$;
8: Initialize $\bar{\boldsymbol{X}}_{N_X} \sim \mathcal{N}\left(\bar{\boldsymbol{X}}_{N_X}; \boldsymbol{0}, \tau_{\max}^2 \boldsymbol{I}\right)$;
9: **for** $j = N_X$ to 1 **do**
10: $\nabla_{\bar{\boldsymbol{X}}_j} \log p\left(\bar{\boldsymbol{X}}_j|\bar{\boldsymbol{Y}}_d\right) = \lambda_x \nabla_{\bar{\boldsymbol{X}}_j} \log p\left(\bar{\boldsymbol{Y}}_d|\bar{\boldsymbol{X}}_j\right) + \nabla_{\bar{\boldsymbol{X}}_j} \log p(\bar{\boldsymbol{X}}_j)$,
11: $\bar{\boldsymbol{X}}_{j-1} = \bar{\boldsymbol{X}}_j + \epsilon_{j,X} \nabla_{\bar{\boldsymbol{X}}_j} \log p(\bar{\boldsymbol{X}}_j|\boldsymbol{Y}_d) + \sqrt{2\epsilon_{j,X}}\boldsymbol{W}$;
12: **end for**
13: Calculate $\boldsymbol{S} = \left[\boldsymbol{P}^{\mathrm{T}}, \boldsymbol{X}_0^{\mathrm{T}}\right]^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$;
14: **end if**
15: **end for**

Ensure: Estimated channel and data $\hat{\boldsymbol{H}} = \boldsymbol{H}_0$ and $\hat{\boldsymbol{X}} = \boldsymbol{X}_0$.

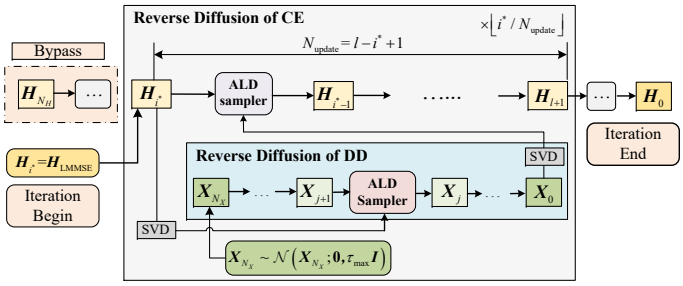


Fig. 1: Block diagram of the proposed diffusion-based asynchronous JCEDD algorithm.

Let $\boldsymbol{R}_\Delta$ represent the estimated covariance matrix of the vectorized channel $\boldsymbol{h} = \mathrm{vec}(\boldsymbol{H}^{\mathrm{T}}) \in \mathbb{C}^{KM}$. Specifically, for a predefined monotonic noise scheduling function $\sigma_t \in [\sigma_{\min}, \sigma_{\max}], t \in [0,1]$, the noise level is discretized into N_H grids for the maximum number of iteration steps. The initial index i^* for the ALD iteration is then determined by identifying σ_i that is closest to the average channel variance, calculated as follows:

$$i^* = \arg\min_{\forall i} \left| \sigma_i - \sum_{l=1}^{KM} [\boldsymbol{R}_\Delta]_{l,l} \right|. \tag{15}$$

- We design an asynchronous alternating iteration mechanism. For channel datasets with complex distributions, the calculation of the posterior score is more complicated than DD that relies solely on model-based iterations with closed-form expressions. Therefore, we update the DD results after every N_{update} channel iterations (Line 6 in Algorithm 1). To ensure the optimal DD results under the current channel error, we conduct N_X complete steps of the ALD sampler each time the data symbols are updated (Lines 9-12).

Additionally, the scaling parameter λ_h and λ_x, as suggested in [20], is empirically chosen to balance the effects of the likelihood score (Line 4) and the prior score (Line 10), enhancing convergence performance.

D. Complexity Analysis

For typical simulation parameters, e.g., $N_{\text{update}} = 50$, $\left\lfloor \frac{i^*}{N_{\text{update}}} \right\rfloor = 5 \sim 15$, $N_X = 1500$ and $N_H = 1500$, the overall complexity of Algorithm 1 is primarily determined by the score function computations in lines 4 and 10. Specifically, the neural network NCSN++ [17], employed to calculate the channel's prior score function, markedly impacts the inference complexity. For complex channel distributions, utilizing a moderately complex network can balance performance with computational demands, providing an avenue for algorithm optimization.

Specifically, computational complexity of calculating each channel likelihood score function is $\mathcal{O}(L^2M + K^2M)$, while the complexity of the channel prior score function is $\mathcal{O}(KF_{\text{net}})$, where F_{net} represents the computational cost of one time inference of the score network $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{h}_t, t)$. The data likelihood score's complexity is $\mathcal{O}(M^2L_d + K^2L_d)$, and the data prior score's complexity is $\mathcal{O}(2KL_d|\mathcal{X}|_c)$.

IV. Simulation Results

A. Simulation Setup

We consider a typical uplink transmission scenario with a total of $K = 12$ users. The BS employs a UPA with $M_x = M_y = 8$ antennas on each dimension, while each user operates with a single antenna. Each element of the pilot sequence $\boldsymbol{p}_k$, $\forall k$, and the data symbols $\boldsymbol{x}_k$ are randomly sampled from 4QAM constellations. We consider a 3GPP-compliant channel model generated using Quadriga [21], simulating the 3GPP 38.901 UMa NLOS scenario at a carrier frequency of 3.5 GHz.

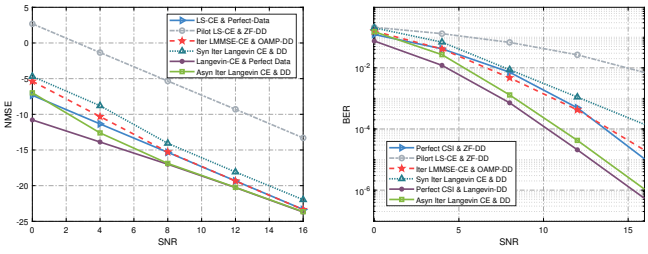
1) Channel Prior Score Network Setup: The NSCN++ architecture [17] is employed to model the prior score function $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{h}_t, t)$. A 3GPP training dataset consisting of 160,000 channel samples is utilized. The time step t is uniformly sampled from the interval $[0, 1]$, and the perturbation noise variance is determined using $\sigma_t = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t$, with $\sigma_{\max} = 30$ and $\sigma_{\min} = 0.01$. The network is trained with a learning rate of 10^{-5} and a batch size of 64, over 20 epochs.

During the testing phase, the reverse diffusion process for CE is discretized into $N_H = 1,500$ steps. Given that an LMMSE initialization is employed, only i^* of the N_H steps are executed. The reverse diffusion process for DD is also discretized into $N_X = 1,500$ steps, using a noise scheduling function defined as $\tau(t) = \tau_{\min}\left(\frac{\tau_{\max}}{\tau_{\min}}\right)^t$, where $\tau_{\max} = 1$ and $\tau_{\min} = 0.01$. Additionally, the hyperparameters are set to $\lambda_h = \lambda_x = 2.5$.

2) Evaluation Metrics: We utilize normalized mean square error (NMSE) and bit error rate (BER) to evaluate the performance of CE and DD, respectively. These metrics are defined respectively as $\text{NMSE} = \mathbb{E}\left\{\frac{\|\boldsymbol{H}-\hat{\boldsymbol{H}}\|^2}{\|\boldsymbol{H}\|^2}\right\}$ and $\text{BER} = \mathbb{E}\left\{\frac{E_b}{KL_d \log_2|\mathcal{X}|_c}\right\}$, where E_b denotes the total number of error bits for active UEs.

3) Comparative Schemes: For CE and DD, the following algorithms are included in the comparisons:

- **LS-CE & Perfect-Data:** The data $\boldsymbol{X}$ is perfectly known, and jointly utilize pilots $\boldsymbol{P}$ and data $\boldsymbol{X}$ for LS CE;
- **Perfect-CSI & ZF-DD:** The CSI $\boldsymbol{H}$ is perfectly known, and employ ZF for DD;
- **Pilot LS-CE & ZF-DD:** Utilize pilots to perform LS CE, and use the estimated channel for ZF DD;
- **Iter LMMSE-CE & OAMP-DD:** Employ LMMSE for CE and OAMP for DD, with CE and DD iteratively executed to enhance the performance of each other [5];
- **Syn Iter Langevin CE & DD:** Conduct synchronous iterative CE and DD using ALD [15], where each channel iteration is followed by one data iteration.
- **Langevin CE & Perfect-Data:** The data $\boldsymbol{X}$ is perfectly known, and apply ALD solver for CE;
- **Perfect-CSI & Langevin-DD:** The CSI $\boldsymbol{H}$ is known, and apply ALD solver for DD;
- **Asyn Iter Langevin CE & DD:** The proposed asynchronous iterative CE and DD using ALD sampler in Algorithm 1.



(a) NMSE (b) BER

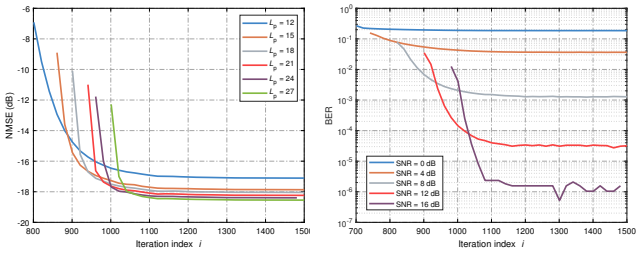
Fig. 2: NMSE and BER performance with respect to the received SNR for different CE and DD algorithms under 3GPP channels.

B. Performance of JCEDD

Fig. 2 illustrates the performance curves of CE and DD under varying SNRs for different schemes. In the simulations, each user is allocated a pilot length of $L_p = 15$ and a data length of $L_d = 50$. As shown in Fig. 2(a), the iterative CE and DD schemes (i.e., **Syn Iter Langevin CE & DD**, **Iter LMMSE-CE & OAMP-DD**, and our **Asyn Iter Langevin CE & DD**) achieve an NMSE gain of approximately 7 and 10 dB when compared to the non-iterative **Pilot LS-CE & ZF-DD** scheme. This indicates a significant gain in CE performance by using detected data to refine CE. As SNR increases, **Iter LMMSE-CE & OAMP-DD**'s performance approaches that of **LS-CE & Perfect-Data**. Similarly, the performance of our **Asyn Iter Langevin CE & DD** matches **Langevin-CE & Perfect-Data** for $\text{SNR} \geq 8$ dB. When the data is not perfectly known, our **Asyn Iter Langevin CE & DD** outperforms **Iter LMMSE-CE & OAMP-DD** because it more effectively utilizes the prior distribution of the channel through a learned prior score function, whereas **Iter LMMSE-CE & OAMP-DD** relies only on the channel's second-order statistical information via LMMSE. Additionally, our **Asyn Iter Langevin CE & DD** implements an asynchronous iteration method to enhance ALD solution performance beyond what **Syn Iter Langevin CE & DD** algorithm can achieve with only a synchronous method.

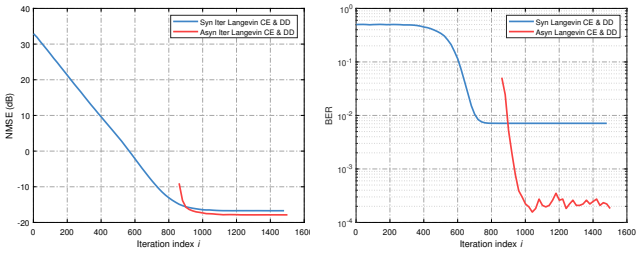
Fig. 2(b) compares the BER performance of various algorithms. Similar to Fig. 2(a), the non-iterative **Pilot LS-CE & ZF-DD** algorithm performs worse than the iterative algorithms. Our **Asyn Iter Langevin CE & DD** scheme achieves performance close to that of **Perfect-CSI & Langevin-DD** and surpasses the **Iter LMMSE-CE & OAMP-DD** and **Syn Iter Langevin CE & DD** algorithms. At a BER of 10^{-4}, our **Asyn Iter Langevin CE & DD** offers a 3 dB SNR gain over **Iter LMMSE-CE & OAMP-DD**. Furthermore, **Perfect-CSI & Langevin-DD** outperforms **Perfect-CSI & ZF-DD**. This is because the ZF algorithm does not exploit the prior distribution information of the constellation. In contrast, **Perfect-CSI & Langevin-DD** effectively leverages the prior distribution of the data and the likelihood information of the received signals, resulting in superior performance.

Fig. 3(a) illustrates the NMSE performance curve of our **Asyn Iter Langevin CE & DD** relative to the iteration index i with varying pilot lengths L_p. The maximum number of iterations is set to $N_H = 1500$ ($i \leq 1500$). The simulation



(a) (b)

Fig. 3: Convergence performance of our Asyn Iter Langevin CE & DD algorithm is evaluated in terms of (a) NMSE with varying pilot lengths L_p and (b) BER under different SNRs, both as function of the iteration index i.



(a) (b)

Fig. 4: NMSE and BER convergence performance of different iteration schemes.

results indicate that as the pilot length L_p increases, the convergence iteration count for the proposed algorithm decreases from approximately 400 iterations at $L_p = 12$ to about 100 iterations at $L_p = 27$. Additionally, the final converged NMSE improves from -17.1 dB to -18.5 dB. Likewise, Fig. 3(b) demonstrates that with different SNRs, both the convergence speed and the final DD outcome of our algorithm improve as the SNR increases. Despite employing a maximum of 1500 iterations, the simulation results reveal that under favorable conditions, such as $L_p = 27$ and SNR $= 16$ dB, the proposed scheme requires only about 50 to 100 iterations to achieve satisfactory results.

Fig. 4 compares the convergence performance of **Syn Iter Langevin CE & DD** and our **Asyn Iter Langevin CE & DD**. Simulation results in Fig. 4 indicate that the **Syn Iter Langevin CE & DD** converges to a local optimum after approximately $i = 800$ iterations for DD and around $i = 1200$ iterations for CE, hindering further improvements in CE and DD for each other. In contrast, the proposed scheme continually enhances BER performance and further improves NMSE performance iteratively. This difference may be due to the discrete distribution of the data, which, in the synchronous scheme without resampling data symbol when using a more accurate channel, can be more easily trapped in the local optimum. These results demonstrate the effectiveness of the proposed asynchronous update strategy, achieving better performance by increasing computational complexity.

V. Conclusion

JCEDD is a core challenge in uplink multiuser transmission for massive MIMO systems, as it heavily determines the QoS for 6G massive communication. Therefore, in this paper, we introduced a generative diffusion model-driven JCEDD scheme utilizing ALD samplers, which significantly improves channel estimation and data detection performance through an asynchronous iterative algorithm. These proposed schemes demonstrate the potential of using generative models to tackle signal detection and estimation challenges. A promising future research direction involves accelerating generative model methods to further reduce inference latency.

References

[1] Z. Gao, *et al.*, "Compressive-sensing-based grant-free massive access for 6G massive communication," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 7411–7435, Mar. 2024.

[2] M. Ke, *et al.*, "Compressive sensing-based adaptive active user detection and channel estimation: Massive access meets massive MIMO," *IEEE Trans. Signal Process.*, vol. 68, pp. 764–779, Feb. 2020.

[3] J. Céspedes, P. M. Olmos, M. Sánchez-Fernández, and F. Perez-Cruz, "Expectation propagation detection for high-order high-dimensional MIMO systems," *IEEE Trans. Commun.*, vol. 62, no. 8, pp. 2840–2849, Aug. 2014.

[4] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Generative diffusion models for high dimensional channel estimation," *IEEE Wireless Commun. Lett.* vol. 7, no. 5, pp. 852–855, Oct. 2018.

[5] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, Mar. 2020.

[6] Y. Cui, S. Li, and W. Zhang, "Jointly sparse signal recovery and support recovery via deep learning with applications in MIMO-based grant-free random access," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 788–803, Mar. 2021.

[7] A. Kosasih, *et al.*, "Graph neural network aided expectation propagation detector for MU-MIMO systems," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2540–2555, Sep. 2022.

[8] K. Pratik, B. D. Rao, and M. Welling, "RE-MIMO: Recurrent and permutation equivariant neural MIMO detection," *IEEE Trans. Signal Process.*, vol. 69, pp. 459–473, Jan. 2021.

[9] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Y, "Diffusion posterior sampling for general noisy inverse problems," in *Proc. Int. Conf. Learn. Represent.*, Feb, 2023.

[10] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 8-14, 2019, pp. 11918–11930.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Adv. Neural Inf. Process. Syst.*, Dec. 6-12, 2020, pp. 6840–6851.

[12] M. Arvinte and J. I. Tamir, "MIMO channel estimation using score-based generative models," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 3698–3713, Jun. 2023.

[13] N. Zilberstein, *et al.*, "Annealed Langevin dynamics for massive MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 3762–3776, Jun. 2023.

[14] N. Zilberstein, A. Sabharwal, and S. Segarra, "Solving linear inverse problems using higher-order annealed Langevin diffusion," *IEEE Trans. Signal Process.*, vol. 72, pp. 492–505, Jan. 2024.

[15] N. Zilberstein, A. Swami, and S. Segarra, "Joint channel estimation and data detection in massive MIMO systems based on diffusion models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Seoul, South Korea, Apr. 14-19, 2024, pp. 13291–13295.

[16] X. Zhou, *et al.*, "Generative diffusion models for high dimensional channel estimation," *IEEE Trans. Wireless Commun.* vol. 24, no. 7, pp. 5840–5854, Jul. 2025.

[17] Y. Song, *et al.*, "Score-based generative modeling through stochastic differential equations," in *Proc. Int. Conf. Learn. Represent.*, May 3-7, 2021, pp. 1–36.

[18] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, Jul. 2011.

[19] B. Efron, "Tweedie's formula and selection bias," *J. Am. Stat. Assoc.*, vol. 106, no. 496, pp. 1602–1614, Jan. 2011.

[20] X. Meng and Y. Kabashima, "Quantized compressed sensing with score-based generative models," in *Proc. Int. Conf. Learn. Represent.*, Kigali, Rwanda, May 1-5, 2023, pp. 1-30.

[21] S. Jaeckel, *et al.*, "QuaDRiGa-quasi deterministic radio channel generator, user manual and documentation," Heinrich Hertz Institute, Tech. Rep. v2.6.1, 2021.