

# Recursive Flow: A Generative Framework for MIMO Channel Estimation

Zehua Jiang<sup>✉</sup>, Fenghao Zhu<sup>✉</sup>, Chongwen Huang<sup>✉</sup>, Richeng Jin<sup>✉</sup>, Zhaohui Yang<sup>✉</sup>,  
Xiaoming Chen<sup>✉</sup>, *Senior Member, IEEE*, Zhaoyang Zhang<sup>✉</sup>, *Senior Member, IEEE*, and  
Mérouane Debbah<sup>✉</sup>, *Fellow, IEEE*

**Abstract**—Channel estimation is a fundamental challenge in massive multiple-input multiple-output systems, where estimation accuracy governs the spectral efficiency and link reliability. In this work, we introduce Recursive Flow (RC-Flow), a novel solver that leverages pre-trained flow matching priors to robustly recover channel state information from noisy, under-determined measurements. Different from conventional open-loop generative models, our approach establishes a closed-loop refinement framework via a serial restart mechanism and anchored trajectory rectification. By synergizing flow-consistent prior directions with data-fidelity proximal projections, the proposed RC-Flow achieves robust channel reconstruction and delivers state-of-the-art performance across diverse noise levels, particularly in noise-dominated scenarios. The framework is further augmented by an adaptive dual-scheduling strategy, offering flexible management of the trade-off between convergence speed and reconstruction accuracy. Theoretically, we analyze the Jacobian spectral radius of the recursive operator to prove its global asymptotic stability. Numerical results demonstrate that RC-Flow reduces inference latency by two orders of magnitude while achieving a 2.7 dB performance gain in low signal-to-noise ratio regimes compared to the score-based baseline.

**Index Terms**—Deep learning, fixed-point iteration, flow matching, generative models, MIMO channel estimation.

## I. INTRODUCTION

The emergence of sixth-generation (6G) wireless networks promises to deliver hyper-connectivity and terabits-per-second data rates [1, 2], with massive multiple-input multiple-output (MIMO) serving as a fundamental enabler to unlock spectral efficiency gains [3, 4]. By deploying large-scale antenna arrays, future base stations can exploit high-resolution spatial multiplexing to serve massive numbers of users simultaneously [5]. However, the theoretical benefits of such massive arrays are fundamentally predicated on the availability of precise channel state information (CSI) at the transmitter [6]. As the number of antennas scales to hundreds or even thousands, the conventional strategy of orthogonal pilot transmission incurs a prohibitive signaling overhead that consumes valuable coherence resources, thereby severely degrading the effective system throughput [7, 8]. Consequently, acquiring high-fidelity CSI from a limited number of pilot measurements has evolved into a severely ill-posed linear inverse problem [9], representing a

critical bottleneck that necessitates algorithmic breakthroughs beyond traditional estimation paradigms.

To mitigate the ill-posedness of channel estimation, conventional approaches resort to statistical or structural regularization to constrain the solution space. The linear minimum mean square error (LMMSE) estimator, widely regarded as the performance benchmark, optimally minimizes the estimation error for Gaussian channels by leveraging second-order statistics [10]. However, its practical deployment is often hindered by the difficulty of acquiring accurate channel covariance matrices, particularly in highly dynamic environments where channel statistics vary rapidly. Alternatively, compressed sensing (CS) techniques have emerged as a powerful paradigm [9], exploiting the sparsity of millimeter-wave channels in the angular domain to recover signals from undersampled measurements. Algorithms such as approximate message passing (AMP) [8] and atomic norm minimization [11] have demonstrated theoretical guarantees under strict sparsity conditions. Nevertheless, these model-based methods fundamentally hinge on ideal assumptions. In realistic scenarios characterized by grid mismatch, energy leakage, or rich scattering, the sparsity assumption is frequently violated, leading to significant performance degradation and reconstruction artifacts.

In order to circumvent the reliance on rigid analytical priors, the research community has pivoted towards data-driven deep learning paradigms, which learn the complex inverse mapping directly from training data. Seminal works have employed convolutional neural networks to estimate channel matrices by exploiting spatial correlations [12, 13]. Furthermore, deep unfolding networks, such as Learned D-AMP [14], unfold iterative algorithms into trainable layers, aiming to combine the interpretability of model-based methods with the expressivity of neural networks. Despite their computational efficiency, these discriminative approaches typically minimize a pixel-wise mean squared error (MSE) loss. Mathematically, such objectives drive the network to approximate the conditional mean of the posterior distribution. While minimizing error variance, this ‘regression to the mean’ behavior inevitably leads to over-smoothed reconstructions [15], resulting in the loss of critical high-frequency spatial details and texture information that are essential for high-precision beamforming.

Recently, generative artificial intelligence has empowered physical layer modeling to transcend the constraints of deterministic regression, enabling a more faithful capture of the inherent stochastic complexity of wireless channels [16–19]. Unlike discriminative models that collapse the posterior

Z. Jiang, F. Zhu, C. Huang, R. Jin, Z. Yang, X. Chen and Z. Zhang are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (E-mails: {jiangzehua, zjuzfh, chongwen-huang, richengjin, yang\_zhaohui, chen\_xiaoming, ning\_ming}@zju.edu.cn).

M. Debbah is with KU 6G Research Center, Department of Computer and Information Engineering, Khalifa University, Abu Dhabi 127788, UAE (E-mail: merouane.debbah@ku.ac.ae).

into a single point estimate, generative approaches explicitly learn the underlying data distribution by reversing a gradual corruption process. Pioneering frameworks, such as denoising diffusion probabilistic models [20] and score-based generative models [21], estimate the score function to iteratively refine Gaussian noise into high-fidelity channel realizations. These methods have been applied in channel synthesis [22] and semantic communications [23], demonstrating superior capability in modeling high-frequency spatial features. Building upon these diffusion foundations, flow matching (FM) has emerged as a cutting-edge generative paradigm [24–27]. By regressing a time-dependent vector field approximating the optimal transport path between prior and data distributions [28], FM rectifies generative trajectories, enabling faster and more deterministic sampling than stochastic diffusion. In the broader context for linear inverse problems, general-purpose solvers such as FLOWER [29] and PnP-Flow [30] have been proposed. Treating the pre-trained flow as a generative prior, these algorithms alternate flow-based denoising and measurement-consistency projections to steer trajectories toward the data manifold and measurement subspace intersection.

Despite the theoretical potential of generative models, their practical deployment for high-dimensional MIMO channel estimation is hindered by specific implementation constraints. A prominent branch of recent flow-based estimators focuses on architectural innovations to accelerate inference. Specifically, the approach in [31] applies standard flow matching to model the conditional channel distribution, while [32] introduces a signal-to-noise ratio (SNR) aware truncation strategy for the reverse diffusion process to reduce computational complexity. Furthermore, [33] leverages the average velocity field to collapse the integration trajectory, enabling channel estimation in a single step. While algorithmically distinct, these methods typically rely on the conjugate transpose of pilot matrices, implicitly assuming pilot orthogonality. In under-determined massive MIMO systems where pilots are limited, this assumption fails, causing severe aliasing artifacts.

Alternatively, the foundational score-based method [34] provides a theoretically rigorous solution for such under-determined systems by employing annealed Langevin dynamics to sample from the posterior. However, the stochastic nature of the dynamics necessitates the continuous injection of noise to correct trajectory errors, requiring thousands of iterative steps that result in prohibitive computational latency. Aiming to reduce this latency, subsequent accelerated deterministic solvers have been proposed. For instance, [35] derives a closed-form approximation of the likelihood score to guide the diffusion reverse process, while [36, 37] utilize Tweedie’s formula [38, 39] or energy-based variational inference to approximate the posterior mean in fewer steps. Nevertheless, critical limitations persist across these deterministic modifications. Their data-consistency updates typically rely on explicit gradient descent steps, where determining the optimal step size presents a significant challenge. Besides, despite the reduction in sampling steps compared to score-based methods [34], these solvers still require hundreds of iterations, which remains computationally prohibitive for low-latency systems. More fundamentally, they operate in an open-loop manner, where the

generative trajectory is initialized once from noise and refined sequentially. In low SNR regimes, the guidance provided by the likelihood term is dominated by heavy measurement noise rather than the true signal residual, causing the solver to drift towards a biased solution.

To resolve the systematic bias inherent in open-loop solvers and the instability of gradient-based physical updates, a paradigm shift is required from ‘trajectory sampling’ to ‘equilibrium finding’. Drawing inspiration from deep equilibrium models [40], which define the output of a network as the fixed point of a non-linear transformation, we propose to reformulate the generative channel estimation process as a closed-loop iteration. The main contributions of this paper can be summarized as follows:

- We introduce Recursive Flow (RC-Flow), a novel generative framework for high-dimensional MIMO channel estimation. By reformulating the inference process as a closed-loop fixed-point iteration and leveraging anchored trajectory rectification, we theoretically guarantee global asymptotic stability towards a fixed point, providing a robust mathematical foundation for generative channel recovery.
- We propose a closed-form proximal projection mechanism to replace heuristic gradient-based updates. This approach ensures unconditional data consistency throughout the iterative process and eliminates the need for tedious step-size tuning, thereby enhancing the solver’s robustness against measurement noise compared to standard diffusion baselines.
- We design an adaptive dual-scheduling strategy  $(\lambda, \beta)$  that dynamically reallocates the computational budget during inference. This mechanism allows the solver to adaptively trade off between convergence speed and restoration precision, enabling the system to meet stringent low-latency requirements essential for real-time 6G applications.
- Extensive simulations demonstrate that RC-Flow achieves SOTA performance across diverse scenarios. In low-SNR regimes, it effectively suppresses noise-induced hallucinations, significantly outperforming existing baselines. In high-SNR regimes, it matches the accuracy of computationally expensive methods while reducing inference latency by orders of magnitude, validating its practical superiority. The code for this paper is available online in [41].

The paper is organized as follows: Section II presents an introduction to the system model and flow matching. Section III introduces the proposed RC-Flow framework, followed by the presentation of simulation results in Section IV. Finally, Section V concludes the paper.

*Notation:* Fonts  $a$ ,  $\mathbf{a}$ , and  $\mathbf{A}$  denote scalars, vectors, and matrices, respectively.  $\|\mathbf{a}\|_2$  represent the 2-norm of  $\mathbf{a}$ .  $\mathbf{A}^T$ ,  $\mathbf{A}^H$ ,  $\mathbf{A}^{-1}$ ,  $\|\mathbf{A}\|_2$  and  $\|\mathbf{A}\|_F$  represent the transpose, conjugate transpose, inverse, 2-norm and Frobenius norm of  $\mathbf{A}$ , respectively. The  $(m, n)$ -th entry of  $\mathbf{A}$  is denoted by  $a_{mn}$ , and  $|\cdot|$  denotes the modulus.  $\mathcal{O}$  represents the asymptotic time complexity. Finally, we can represent the diagonal matrix and trace of matrix  $\mathbf{A}$  using the notations  $\text{diag}(\mathbf{A})$  and  $\text{Tr}(\mathbf{A})$ , respectively.

## II. PRELIMINARIES

### A. Wireless System Model

We consider a narrowband, point-to-point MIMO communication system consisting of  $N_t$  transmit antennas and  $N_r$  receive antennas. The wireless propagation environment is characterized by the complex-valued CSI matrix  $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ . To estimate this channel, the transmitter sends a sequence of  $N_p$  pilot symbols. Let  $\mathbf{p}_i \in \mathbb{C}^{N_t}$  denote the  $i$ -th pilot symbol and  $\mathbf{W} \in \mathbb{C}^{N_r \times N_r}$  be the receive beamforming matrix. The received signal vector for the  $i$ -th pilot  $\mathbf{y}_i$  is given by:

$$\mathbf{y}_i = \mathbf{W}^H (\mathbf{H} \mathbf{p}_i s + \mathbf{n}_i), \quad (1)$$

where  $\mathbf{n}_i$  is the complex additive white Gaussian noise and  $s$  is a complex-valued scalar. For the remainder of the paper, we assume  $s = 1$  and a fully digital receiver where  $\mathbf{W} = \mathbf{I}$ . The pilot  $\mathbf{p}_i$  is constrained to have unit-amplitude entries with low-resolution phase, where each element is a randomly selected quadrature phase shift keying symbol that remains fixed for all test samples. Assuming the channel remains constant over the duration of the pilot transmission, the received signal matrix  $\mathbf{Y}$  is modeled as:

$$\mathbf{Y} = \mathbf{H} \mathbf{P} + \mathbf{N}, \quad (2)$$

where  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{N_p}] \in \mathbb{C}^{N_r \times N_p}$  is the received signal matrix,  $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{N_p}] \in \mathbb{C}^{N_t \times N_p}$  is the pilot matrix,  $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_{N_p}] \in \mathbb{C}^{N_r \times N_p}$  is the additive noise matrix. The entries of  $\mathbf{N}$  are typically assumed to be independent and identically distributed complex white Gaussian noise with zero mean and variance  $\sigma_{\text{pilot}}^2$ .

The objective of channel estimation is to recover the high-dimensional matrix  $\mathbf{H}$  from the observed measurements  $\mathbf{Y}$ , given the knowledge of the pilot matrix  $\mathbf{P}$ . We define the pilot density as  $\alpha = N_p/N_t$ . In practical mmWave scenarios, pilot resources are often severely constrained ( $\alpha < 1$ ), thereby rendering the recovery of an inherently under-determined inverse problem.

### B. Continuous Normalizing Flows

Continuous normalizing flows represent a class of generative models that transform a simple prior distribution  $p_0(\mathbf{H})$  (e.g., a standard Gaussian distribution) into a complex data distribution  $p_1(\mathbf{H})$  through a continuous-time generative process [25, 26]. This process is characterized by a flow, denoted as:

$$\phi_t : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d, \quad (3)$$

where  $d$  represents the dimension of the data, the symbol  $\times$  and  $\rightarrow$  denote the Cartesian Product and the mapping between data domains, respectively. Intuitively,  $\phi_t$  moves a sample from its initial state at  $t = 0$  to a target state at  $t = 1$ . Mathematically, for any time  $t$  and any channel matrix  $\mathbf{H}$  in the signal space, the flow is defined by an ordinary differential equation (ODE):

$$\frac{d}{dt} \phi_t(\mathbf{H}) = \mathbf{V}_t(\phi_t(\mathbf{H})), \quad \phi_0(\mathbf{H}) = \mathbf{H}_0, \quad (4)$$

where  $\phi_0(\mathbf{H}) = \mathbf{H}_0$  defines the boundary condition,  $\mathbf{V}_t$  is the time-dependent velocity field that determines the instantaneous direction and speed of the evolution. This sample-wise

transport induces a time-dependent probability density path  $p_t$ . The coupling between the density  $p_t$  and the velocity field  $\mathbf{V}_t$  is formally described by the continuity equation:

$$\frac{\partial p_t(\mathbf{H})}{\partial t} + \text{div}(p_t(\mathbf{H}) \mathbf{V}_t(\mathbf{H})) = 0, \quad (5)$$

which ensures that the velocity field correctly describes the evolution of the probability density over time.

### C. Flow Matching

The objective of flow matching (FM) is to train a deep neural network, parameterized by  $\theta$ , to approximate the optimal marginal velocity field  $\mathbf{V}_t$  [24, 30, 31]. The network parameterizes a time-dependent vector field  $\mathbf{V}_t^\theta$ , which is optimized to minimize the MSE-based FM loss:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}[0,1], \mathbf{H} \sim p_t(\mathbf{H})} [\|\mathbf{V}_t^\theta(\mathbf{H}) - \mathbf{V}_t(\mathbf{H})\|_F^2]. \quad (6)$$

However, directly minimizing  $\mathcal{L}_{\text{FM}}$  is typically infeasible in practice as both the marginal density  $p_t(\mathbf{H})$  and the target field  $\mathbf{V}_t$  are generally inaccessible. To circumvent this, conditional flow matching (CFM) is utilized [42], which simplifies the optimization by conditioning on individual data samples  $\mathbf{H}_1$  drawn from the empirical distribution  $p_1(\mathbf{H})$ . By defining a conditional probability path  $p_t(\mathbf{H}|\mathbf{H}_1)$  and a corresponding conditional velocity field  $\mathbf{V}_t(\mathbf{H}|\mathbf{H}_1)$  that transports probability mass toward  $\mathbf{H}_1$ , the CFM objective is formulated as:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, \mathbf{H}_1 \sim p_1, \mathbf{H} \sim p_t(\mathbf{H}|\mathbf{H}_1)} [\|\mathbf{V}_t^\theta(\mathbf{H}) - \mathbf{V}_t(\mathbf{H}|\mathbf{H}_1)\|_F^2]. \quad (7)$$

Notably, the gradients of  $\mathcal{L}_{\text{CFM}}$  and  $\mathcal{L}_{\text{FM}}$  with respect to  $\theta$  are equivalent, provided that the marginal velocity field is the aggregation of its conditional counterparts:

$$\mathbf{V}_t(\mathbf{H}) = \int \mathbf{V}_t(\mathbf{H}|\mathbf{H}_1) \frac{p_t(\mathbf{H}|\mathbf{H}_1) p_1(\mathbf{H}_1)}{p_t(\mathbf{H})} d\mathbf{H}_1. \quad (8)$$

This framework enables ‘simulation-free’ training. Specifically, for Gaussian conditional paths, samples at any time  $t$  can be generated via a closed-form reparameterization, eliminating the need for expensive ODE solvers during the training phase.

In our implementation, we adopt a Gaussian conditional probability path  $p_t(\mathbf{H}|\mathbf{H}_1) = \mathcal{N}(\mathbf{H}; \mu_t(\mathbf{H}_1), \sigma_t^2 \mathbf{I})$  and employ a linear interpolation strategy to construct the flow. This trajectory is defined by the displacement map:

$$\psi_t(\mathbf{H}_0, \mathbf{H}_1) = t\mathbf{H}_1 + (1-t)\mathbf{H}_0, \quad (9)$$

where  $\mathbf{H}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  denotes the initial noise distribution, and the mean and variance evolve as  $\mu_t(\mathbf{H}_1) = t\mathbf{H}_1$  and  $\sigma_t^2 = (1-t)^2$ , respectively. Such a construction induces a time-independent conditional velocity field:

$$\mathbf{V}_t(\mathbf{H}|\mathbf{H}_1) = \frac{d}{dt} \psi_t(\mathbf{H}_0, \mathbf{H}_1) = \mathbf{H}_1 - \mathbf{H}_0. \quad (10)$$

Compared to curved diffusion-based paths, this rectified linear trajectory significantly accelerates training convergence and enhances sampling efficiency during inference. Finally, to align with the estimation task, we define the generative process via a reverse-time flow, evolving from  $t = 1$  back to  $t = 0$  (ground-truth).

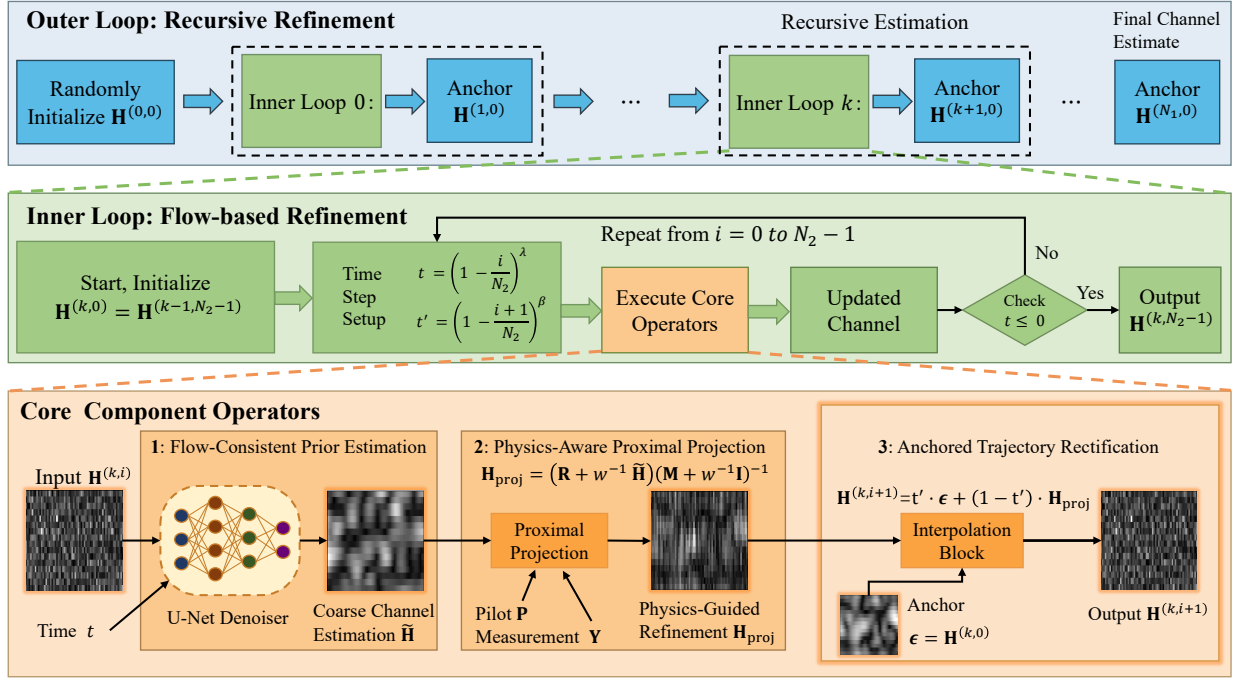


Fig. 1. Schematic of the RC-Flow algorithm. The method employs a nested loop structure: the Outer Loop recursively resets the anchor to correct trajectory drift, while the Inner Loop refines the estimate. The Core Component Operators combine a deep flow prior, physics-aware proximal projection, and anchored rectification to achieve robust reconstruction.

### III. RECURSIVE FLOW FRAMEWORK

In this section, we present the RC-Flow framework, as illustrated in Fig. 1 and summarized in Algorithm 1. We structure the presentation hierarchically: first, we define the core component operators that serve as the fundamental building blocks; second, we detail the inner loop mechanism driven by an adaptive dual-scheduling strategy; third, we introduce the outer loop recursive anchor refinement for trajectory rectification; finally, we provide a rigorous theoretical analysis establishing the existence and local asymptotic stability of the solver's fixed point.

#### A. Core Component Operators

*a) Flow-Consistent Prior Estimation:* To establish an informative generative prior for channel estimation, we utilize the CFM framework pre-trained on empirical channel realizations. In the training phase, the network, parameterized by  $\theta$ , learns to approximate a time-dependent velocity field. For each training instance, an SNR is sampled from a predefined discrete set  $\mathcal{S}_{dB}$ . Based on the selected SNR, complex Gaussian noise is generated and superimposed on the ground-truth channel  $\mathbf{H}$ . Here,  $\mathbf{H}$  is globally normalized such that  $\mathbb{E}[|h_{ij}|^2] = 1$ . We define the probability density path through linear interpolation between the noise-free source  $\mathbf{H}_0 = \mathbf{H}$  and the noisy target  $\mathbf{H}_1 = \mathbf{H} + \mathbf{N}$ , expressed as  $\mathbf{H}_t = (1-t)\mathbf{H}_0 + t\mathbf{H}_1$ . To optimize the learning efficiency across the flow trajectory, the temporal variable  $t \in [0, 1]$  follows a logit-normal schedule. The parameters  $\theta$  are optimized by minimizing the MSE between the predicted and target velocity fields:

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \mathbf{H}, \mathbf{N}} [\|\mathbf{V}_t^\theta(\mathbf{H}_t, t) - (\mathbf{H}_1 - \mathbf{H}_0)\|_F^2]. \quad (11)$$

During inference, the objective is to recover the clean channel from a noisy intermediate state by leveraging the learned flow dynamics. Specifically, given a current state  $\mathbf{H}^{(k,i)}$  and a time step  $t$ , the pre-trained network predicts the velocity  $\mathbf{V} = \text{Model}(\mathbf{H}^{(k,i)}, t)$  tangent to the probability flow. A flow-consistent prior estimate  $\tilde{\mathbf{H}}$  is then obtained by projecting the state along the trajectory back to  $t = 0$ . This yields the following deterministic single-step estimation formula:

$$\tilde{\mathbf{H}} = \mathbf{H}^{(k,i)} - t \cdot \mathbf{V}. \quad (12)$$

This operation effectively utilizes the learned generative prior to denoise the channel from the current observation.

*b) Physics-Aware Proximal Projection:* While the prior estimate  $\tilde{\mathbf{H}}$  is consistent with the learned distribution, it may not strictly adhere to the physical constraints imposed by the pilot observations. To enforce data consistency, we employ a proximal operator that reconciles the generative prior with the measurement likelihood. This is formulated as a regularized least-squares problem:

$$\mathbf{H}_{\text{proj}} = \arg \min_{\mathbf{H}} \frac{1}{2\sigma_{\text{pilot}}^2} \|\mathbf{Y} - \mathbf{H}\mathbf{P}\|_F^2 + \frac{1}{2w} \|\mathbf{H} - \tilde{\mathbf{H}}\|_F^2, \quad (13)$$

where  $w$  is a time-dependent regularization parameter. We adopt a variance annealing schedule defined as  $w = t^2/(t^2 + (1-t)^2)$ , which dynamically balances the data fidelity of the pilot measurements with the flow-consistent prior.

As a quadratic optimization problem, (13) admits a unique closed-form solution:

$$\mathbf{H}_{\text{proj}} = (\mathbf{R} + w^{-1}\tilde{\mathbf{H}})(\mathbf{M} + w^{-1}\mathbf{I})^{-1}, \quad (14)$$

where  $\mathbf{M} = \sigma_{\text{pilot}}^{-2} \mathbf{P}\mathbf{P}^H$  and  $\mathbf{R} = \sigma_{\text{pilot}}^{-2} \mathbf{Y}\mathbf{Y}^H$ . To circumvent the prohibitive computational complexity associated with direct matrix inversion, we leverage eigenvalue decomposition

---

**Algorithm 1** Recursive Flow Algorithm

---

**Input:** Observations  $\mathbf{Y}$ , Pilots  $\mathbf{P}$ ,  $\sigma_{\text{pilot}}$ ,  $N_1, N_2, \lambda, \beta$ .  
**Output:** Estimated Channel  $\mathbf{H}_{\text{est}}$ .  
1: Initialize  $\mathbf{M} = \sigma_{\text{pilot}}^{-2} \mathbf{P} \mathbf{P}^H$  and  $\mathbf{R} = \sigma_{\text{pilot}}^{-2} \mathbf{Y} \mathbf{P}^H$ .  
2: EVD:  $\mathbf{M} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H$ .  
3: Initialize  $\mathbf{H}^{(0,0)} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$ .  
4: **for**  $k \leftarrow 0, \dots, N_1 - 1$  **do**  
5:   **for**  $i \leftarrow 0, \dots, N_2 - 1$  **do**  
6:      $t = (1 - i/N_2)^\lambda$   
7:      $t' = (1 - (i+1)/N_2)^\beta$   
8:      $\mathbf{V} = \text{Model}(\mathbf{H}^{(k,i)}, t)$   
9:      $\tilde{\mathbf{H}} = \mathbf{H}^{(k,i)} - t \cdot \mathbf{V}$   $\triangleright$  Denoising  
10:      $w = t^2 / (t^2 + (1 - t)^2)$   
11:      $\mathbf{H}_{\text{proj}} = (\mathbf{R} + w^{-1} \tilde{\mathbf{H}}) \mathbf{U} (\mathbf{\Lambda} + w^{-1} \mathbf{I})^{-1} \mathbf{U}^H$   
12:      $\triangleright$  Projection  
13:     **if**  $k = 0$  **then**  
14:        $\epsilon \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$   $\triangleright$  Initialize Anchor  
15:     **else**  
16:        $\epsilon = \mathbf{H}^{(k-1, N_2-1)}$   $\triangleright$  Reset Anchor  
17:     **end if**  
18:      $\mathbf{H}^{(k, i+1)} = t' \cdot \epsilon + (1 - t') \cdot \mathbf{H}_{\text{proj}}$   $\triangleright$  Interpolation  
19:   **end for**  
20:    $\mathbf{H}^{(k+1, 0)} = \mathbf{H}^{(k, N_2-1)}$   
21: **end for**  
22:  $\mathbf{H}_{\text{est}} = \mathbf{H}^{(N_1-1, N_2-1)}$   
23: **return**  $\mathbf{H}_{\text{est}}$

---

(EVD). Given that  $\mathbf{M}$  is a Hermitian matrix, its EVD is expressed as:

$$\mathbf{M} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H, \quad (15)$$

where  $\mathbf{U}$  is a unitary matrix and  $\mathbf{\Lambda}$  is a diagonal matrix of eigenvalues. Consequently, the regularized inversion term in (14) can be efficiently reformulated as:

$$(\mathbf{M} + w^{-1} \mathbf{I})^{-1} = \mathbf{U} (\mathbf{\Lambda} + w^{-1} \mathbf{I})^{-1} \mathbf{U}^H. \quad (16)$$

Since  $(\mathbf{\Lambda} + w^{-1} \mathbf{I})$  remains diagonal, the inversion is reduced to an element-wise reciprocal of its diagonal entries, significantly reducing the per-iteration computational overhead.

*c) Anchored Trajectory Rectification:* Although the proximal projection enforces data consistency, directly adopting the refined estimate  $\mathbf{H}_{\text{proj}}$  may cause the state to deviate from the learned probability flow trajectory. To mitigate this trajectory drift and maintain structural integrity, we implement an anchored rectification strategy designed to preserve the generative path. Specifically, we define the latent anchor  $\epsilon = \mathbf{H}^{(k,0)}$  to represent the source of the flow. The updated state for the subsequent time step  $t'$  is then reconstructed by re-interpolating between the anchor and the physically refined estimate  $\mathbf{H}_{\text{proj}}$ , expressed as:

$$\mathbf{H}^{(k, i+1)} = t' \cdot \epsilon + (1 - t') \cdot \mathbf{H}_{\text{proj}}. \quad (17)$$

This formulation ensures that the transition from noise to data adheres to the optimal straight-line path, while effectively steering the generative evolution toward the physically consistent manifold.

### B. Inner Loop: Flow-based Refinement

The aforementioned operators form the core processing unit of the proposed framework, integrated into an inner loop designed to progressively refine the channel estimate from a noisy observation toward a ground-truth realization. The flow-based refinement process is structured as follows:

In the  $k$ -th outer iteration, the inner loop is initialized with the anchor  $\mathbf{H}^{(k,0)} = \mathbf{H}^{(k-1, N_2-1)}$  as dictated by the recursive anchor logic. The loop then executes  $N_2$  steps ( $i \in [0, N_2-1]$ ) to traverse the probability flow trajectory.

To balance trajectory stability with the capture of rapid velocity dynamics near the data manifold, we employ an adaptive dual-scheduling strategy. This approach decouples the time coordinate for feature extraction from the temporal step used for noise removal. Specifically, for the  $i$ -th inner step, the current time  $t$  and target time  $t'$  are governed by two distinct polynomial schedules:

$$t = (1 - i/N_2)^\lambda, \quad (18)$$

$$t' = (1 - (i+1)/N_2)^\beta. \quad (19)$$

Where  $\lambda$  modulates the time embedding of the network to govern the granularity of prior estimation, while  $\beta$  regulates the rectification step size and the rate of anchor removal. This mechanism facilitates rapid progression in high-noise regimes while ensuring fine-grained adjustments as the channel structure emerges. The impact of  $\lambda$  and  $\beta$  configurations on convergence speed and estimation performance is further analyzed in Section IV.

Given the time indices  $t$  and  $t'$ , the loop updates the channel state  $\mathbf{H}^{(k,i)}$  by sequentially applying the operators defined in (12)–(17). This sequence integrates flow-consistent prior extraction, physics-aware projection, and trajectory rectification to generate the subsequent state  $\mathbf{H}^{(k, i+1)}$ . This procedure repeats until  $i = N_2 - 1$  (where  $t' = 0$ ). The final state  $\mathbf{H}^{(k, N_2-1)}$  serves as both the refined channel estimate for the current outer iteration and the anchor for the recursive update in the subsequent stage.

### C. Outer Loop: Recursive Anchor Refinement

While the inner loop integrates the core operators, a singular traversal of the probability flow trajectory often fails to obtain an optimal channel estimate. The substantial gap between the Gaussian prior and the observed posterior, coupled with non-linear corrections from proximal projections, typically results in suboptimal convergence. To address this, we propose a recursive anchor refinement strategy. In contrast to standard single-pass inference, this approach leverages the output of the current stage to re-initialize the anchor for the subsequent stage, effectively moving the generative flow source closer to the target data manifold.

Formally, the anchor serves as the source state of the probability path at  $t = 1$ , functioning as the fixed reference point for the trajectory rectification step in (17). In this recursive scheme, the final estimate from the  $k$ -th inner loop, denoted as  $\mathbf{H}^{(k, N_2-1)}$ , is assigned as the anchor for the subsequent iteration. This update rule is formulated as:

$$\mathbf{H}^{(k+1, 0)} = \mathbf{H}^{(k, N_2-1)}, \quad (20)$$

$$\mathbf{H}^{(0,0)} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}). \quad (21)$$

where  $\mathbf{H}^{(0,0)}$  denotes the initial Gaussian noise. This recursion proceeds for  $N_1$  outer iterations, after which the final state  $\mathbf{H}^{(N_1,0)}$  is output as the estimated channel matrix.

This anchor-based refinement functions as a critical self-correction mechanism. By re-initializing the flow source with the most recent estimate, the algorithm effectively compensates for trajectory drift induced by the finite discretization steps of the inner loop. Consequently, this recursive process progressively steers the estimate toward a high-fidelity solution that is consistent with both learned priors and physical measurements.

#### D. Theoretical Analysis: Global Asymptotic Stability

We model the algorithm as a discrete dynamical system governed by the composite operator  $\mathcal{T} : \mathbb{C}^{N_r \times N_t} \rightarrow \mathbb{C}^{N_r \times N_t}$ . Specifically, a single inner-loop iteration can be expressed as:

$$\mathbf{H}^{(k+1,0)} = \mathcal{T}(\mathbf{H}^{(k,0)}) \triangleq \mathcal{A} \circ \mathcal{P} \circ \mathcal{D}(\mathbf{H}^{(k,0)}), \quad (22)$$

where  $\mathcal{D}$  represents the flow-matching denoiser,  $\mathcal{P}$  denotes the physics-aware proximal projection, and  $\mathcal{A}$  is the anchored rectification step.

To prove the existence of a fixed point  $\mathbf{H}^* = \mathcal{T}(\mathbf{H}^*)$ , we adopt the standard assumption of *Bounded Denoisers*, which is widely used in the analysis of Plug-and-Play algorithms [43].

**Assumption 1** (Bounded Denoiser Hypothesis). *The pre-trained flow matching model acts as a bounded denoiser. This indicates that for any input state  $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ , the output magnitude is bounded by a constant  $B_{\text{flow}} < \infty$ :*

$$\|\mathcal{D}(\mathbf{H})\|_F \leq B_{\text{flow}}. \quad (23)$$

Under this mild assumption, we can utilize the properties of the physical projection operator to guarantee the existence of equilibrium states.

**Theorem 1** (Existence of Fixed Point). *Consider the composite operator  $\mathcal{T}$  with anchor coefficient  $t'_i \in (0, 1)$  and regularization weight  $w > 0$ . Under Assumption 1, there exists a compact convex set  $\mathcal{K} \subset \mathbb{C}^{N_r \times N_t}$  such that  $\mathcal{T}$  maps  $\mathcal{K}$  to itself ( $\mathcal{T}(\mathcal{K}) \subseteq \mathcal{K}$ ). Consequently, by Brouwer's Fixed-Point Theorem [44], there is at least one fixed point  $\mathbf{H}^* \in \mathcal{K}$ .*

*Proof.* See Appendix A.  $\square$

Having established the existence of a fixed point, we now analyze the stability of the RC-Flow solver. We define the total Jacobian of the outer-loop operator  $\mathcal{T}$  at the fixed point  $\mathbf{H}^*$  as  $\mathbf{J}_{\mathcal{T}} \triangleq \frac{\partial \text{vec}(\mathbf{H}^{(k+1)})}{\partial \text{vec}(\mathbf{H}^{(k)})} \big|_{\mathbf{H}^*}$ . In high-dimensional inverse problems, generative priors often exhibit local expansion (i.e., Lipschitz constant  $> 1$ ) to recover high-frequency textures. However, our framework relies on the geometric interplay between the prior and the physical constraints. We encapsulate this via the following hypothesis.

**Assumption 2** (Spectral Contraction of Composite Operator). *Let  $\mathbf{J}_{\mathcal{D},i}$  and  $\mathbf{J}_{\mathcal{P},i}$  be the Jacobians of the flow denoiser and the physics-aware projection at the  $i$ -th inner step, respectively. We assume that the projection operator imposes sufficient*

Table I  
SYSTEM CONFIGURATIONS

Parameter	Value
<i>Training Configurations</i>	
Optimizer	Adam
Learning Rate	$1 \times 10^{-4}$
Batch Size	256
Total Epochs	1600
Training SNR Range	$\{-10, -5, \dots, 30\}$
Time Sample	$\sigma(\xi)$
EMA Decay Rate	0.999
Precision	bf16
<i>Inference Configurations</i>	
Time Schedule Parameter ( $\lambda, \beta$ )	2,2
Model Selection (Epoch)	1600
Inner Loop Number	(25)
$N_{\text{max}}, N_{\text{min}}$	50,3
<i>Network Architecture</i>	
Input/Output Channels	2 (Real/Imag)
Base Model Channels	32
Channel Multipliers	(1, 2, 4)
Residual Blocks per Level	2
Time Embedding Dimension	128
Group Normalization Numbers	8
Activation (ResNet, Time MLP)	SiLU, GELU
Downsampling/Upsampling Steps	2

structural constraints to suppress the expansive modes of the denoiser, such that the spectral radius of the composite Jacobian  $\mathbf{T}_i \triangleq \mathbf{J}_{\mathcal{P},i} \mathbf{J}_{\mathcal{D},i}$  satisfies:

$$\rho(\mathbf{T}_i) \leq \gamma < 1. \quad (24)$$

Under this assumption, we have the following theorem:

**Theorem 2** (Global Asymptotic Stability in Generalized Metric Space). *Suppose the inner-loop operators satisfy Assumption 2, and the anchor coefficients  $\{t'_i\}_{i=0}^{N_2-1}$  are time-varying within  $[0, 1)$ . Then there exists a generalized vector norm  $\|\cdot\|_*$  under which the total Jacobian  $\mathbf{J}_{\mathcal{T}}$  is strictly contractive, i.e.,  $\|\mathbf{J}_{\mathcal{T}}\|_* < 1$ . Consequently, the recursive iteration converges linearly to the fixed point  $\mathbf{H}^*$  for any initialization within the basin of attraction.*

*Proof.* See Appendix B.  $\square$

Therefore, due to the equivalence of norms in finite-dimensional spaces, convergence in  $\|\cdot\|_*$  implies global asymptotic stability in the standard Euclidean norm  $\|\cdot\|_2$ .

## IV. NUMERICAL RESULTS

In this section, we provide a comprehensive numerical evaluation of the proposed RC-Flow framework across a wide range of operational conditions. Following the algorithmic established in Section III, the performance and robustness of the scheme are systematically evaluated in the following subsections.

### A. System Configurations

a) *Dataset Generation:* We evaluate the performance of the proposed RC-Flow using the 3GPP clustered delay line (CDL) channel models, which represent realistic propagation

environments for 5G/6G MIMO systems [45]. The dataset comprises four distinct channel profiles: CDL-A, CDL-B, CDL-C, and CDL-D. Specifically, CDL-D represents a line-of-sight (LOS) environment, whereas CDL-B and CDL-C represent non-line-of-sight (NLOS) scenarios. CDL-A contains a mixture of both LOS and NLOS components. For each channel model, we generate 10,000 independent channel realizations for training and 100 realizations for testing. Furthermore, a mixed dataset is constructed by aggregating the training samples from all four CDL profiles.

*b) Network Architecture:* The CFM network employs a time-conditioned U-Net architecture, and the real and imaginary channel components are processed as a two-channel real-valued input. The backbone follows an encoder-decoder structure with multi-level ResNet blocks [46], whose specific configurations are detailed in Table I. Temporal information  $t$  is integrated into each block via sinusoidal positional encoding and a scale-and-shift mechanism.

*c) Training and Inference:* The model is optimized via the CFM objective using the Adam optimizer. To ensure robustness, the training SNR is uniformly sampled from  $S_{dB} \in \{-10, -5, \dots, 30\}$  for each batch. Additionally, the time step  $t$  is sampled from a logit-normal distribution, defined as  $t = \sigma(\xi)$  where  $\xi \sim \mathcal{N}(0, 1)$ . Furthermore, an exponential moving average (EMA) is maintained to stabilize the weights, with comprehensive configurations detailed in Table I.

During inference,  $N_2$  is adaptively scheduled based on the noise standard deviation to balance estimation accuracy and computational complexity, as follows:

$$N_2 = N_{\min} + \Delta N, \quad (25)$$

$$\Delta N = \left\lfloor (N_{\max} - N_{\min}) \cdot \left( \frac{\log_{10}(\sigma_{\max}/\sigma_{\text{pilot}})}{\log_{10}(\sigma_{\max}/\sigma_{\min})} \right)^2 \right\rfloor, \quad (26)$$

where  $N_{\max}$  and  $N_{\min}$  represent the predefined upper and lower bounds of the inner loop iteration numbers, while  $\sigma_{\max}$  and  $\sigma_{\min}$  denote the noise standard deviation boundaries that delineate the dynamic range of the environment.

*d) Hardware and Software Environment:* All experiments were implemented in Python 3.10 using the PyTorch 2.6.0 on the Ubuntu 22.04 system. The training and inference processes were accelerated using an NVIDIA A100 GPU.

## B. Baselines

To validate the effectiveness of the proposed method, we compare it against several baselines, ranging from classical estimation theory to recent deep learning-based approaches. The following baselines are evaluated in the simulations:

- **Linear Minimum Mean Square Error (LMMSE)** [10]: This estimator represents the optimal linear strategy designed to minimize the Bayesian MSE between the estimated and true channel matrices. By incorporating prior statistical knowledge, specifically the channel spatial correlation matrix, it effectively balances the reliability of the received observation against the prior distribution of the channel. The formulation leverages the second-order statistics of the channel, expressed as:

$$\mathbf{H}_{\text{LMMSE}} = \mathbf{Y}(\mathbf{P}^H \mathbf{R}_{\mathbf{H}} \mathbf{P} + \sigma_{\text{pilot}}^2 \mathbf{I})^{-1} \mathbf{P}^H \mathbf{R}_{\mathbf{H}} \quad (27)$$

where  $\mathbf{R}_{\mathbf{H}}$  denotes the channel covariance matrix representing the spatial correlation.

- **Frequency-selective Atomic Decomposition (fsAD)** [11]: Representing the class of CS methods, the fsAD algorithm exploits the inherent sparsity of millimeter-wave channels in the continuous angular domain. It formulates channel estimation as an atomic norm minimization problem. While effective for sparse channels, its performance may degrade in rich scattering environments where the channel exhibits low sparsity or when the number of propagation paths increases significantly, challenging the fundamental CS assumptions.
- **Learned Denoising-based Approximate Message Passing (L-DAMP)** [14]: A model-based deep learning approach that unfolds the Approximate Message Passing (AMP) algorithm into a neural network architecture. By replacing the traditional shrinkage function in AMP with a learned denoiser (typically a DnCNN [15]), L-DAMP effectively learns the channel prior from training data to reconstruct sparse signals from noisy measurements.
- **Score-based** [34]: A generative approach based on the noise conditional score network. This method learns the gradient of the log-prior distribution (the score function) of the channel. During inference, it employs annealed Langevin dynamics to iteratively denoise a random initialization, producing a single sample from the posterior distribution  $p(\mathbf{h}|\mathbf{y})$ . This represents the performance of posterior sampling.
- **Lower Bound** [34]: An enhanced inference strategy utilizing the same pre-trained Score-based model as above [34]. To establish a performance lower bound, this strategy approximates the MMSE estimator using the same pre-trained Score-based model. Instead of relying on a single posterior sample, it computes the posterior mean  $\mathbb{E}[\mathbf{h}|\mathbf{y}]$  by averaging 50 independent samples generated from parallel Langevin dynamics chains, thereby minimizing stochastic variance and providing the theoretically optimal error floor for the score-based approach.

## C. CSI Estimation Evaluation

Fig. 2 illustrates the robustness of the proposed RC-Flow in CDL- $\{A, B, C, D\}$  environments, employing a  $16 \times 64$  MIMO configuration with  $\alpha = 0.6$ . For comparison, a Mixed RC-Flow model is trained on a composite dataset of CDL- $\{A, B, C, D\}$  comprising 40,000 channel realizations. Furthermore, the performance of in-distribution models trained specifically on each corresponding CDL model is included to characterize the estimation accuracy within matched environments. As for performance measurement, we use normalized mean squared error (NMSE), defined as:

$$\text{NMSE} = 10 \log_{10} \frac{\|\mathbf{H}_{\text{est}} - \mathbf{H}\|_F^2}{\|\mathbf{H}\|_F^2} \quad (28)$$

In in-distribution scenarios, the proposed RC-Flow outperforms the score-based baseline and closely approaches the lower bound. While RC-Flow exhibits nearly identical precision to the score-based method in high-SNR regimes, it achieves a substantial performance gain of approximately



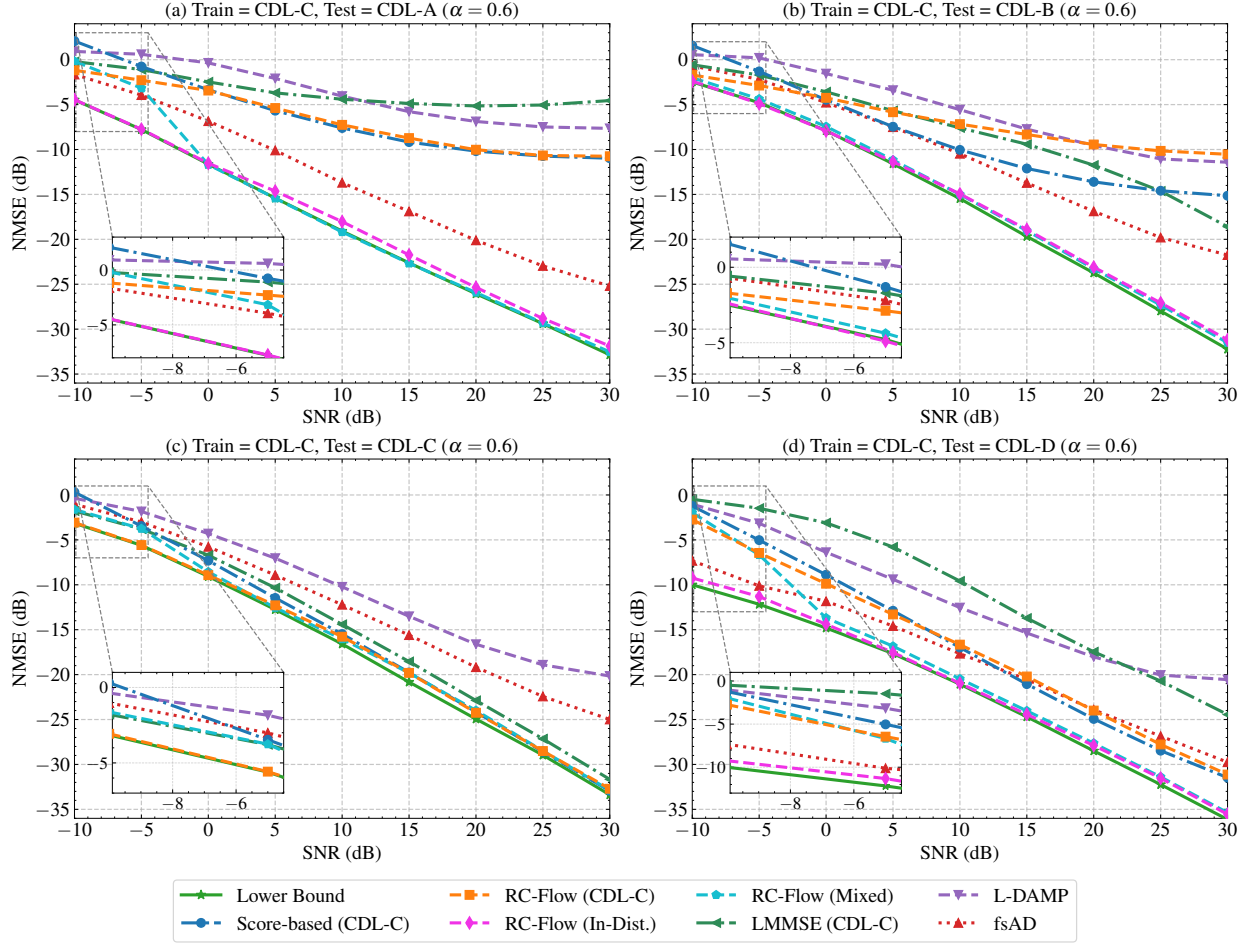


Fig. 2. Performance comparison of various schemes trained on CDL-C channels and evaluated across diverse channel conditions in a  $16 \times 64$  mmWave MIMO system when  $\alpha = 0.6$ .

2 dB in the low-SNR region ( $-10$  dB to  $-5$  dB). This superior noise resilience underscores the potential of RC-Flow as a high-fidelity estimation framework, particularly for power-limited environments. Although the LMMSE estimator is competitive across all SNR regimes, it consistently lags behind the proposed RC-Flow by a margin exceeding 2 dB.

The generalization capabilities of the generative priors are further analyzed through the Mixed model and out-of-distribution (OOD) evaluations. The Mixed model maintains robustness at high SNR but suffers pronounced NMSE degradation at low SNR (excluding CDL-B) relative to its in-distribution counterparts. This stems from the structural ambiguity of a generalized prior, which lacks the specialization necessary to resolve profile-specific features when measurement guidance is weak. In OOD scenarios, the RC-Flow trained on CDL-C exhibits an NMSE floor of approximately  $-11$  dB when evaluated on CDL-A and CDL-B. In contrast, negligible saturation is observed on CDL-D channels, where the NMSE decreases consistently with SNR, maintaining a 5 dB gap relative to the in-distribution benchmark. This discrepancy is attributed to channel complexity: the rich scattering in CDL-A and B increases estimation difficulty, whereas the dominant LOS components in CDL-D facilitate simpler recovery.

Finally, conventional baselines such as L-DAMP and fsAD

provide relatively reliable and consistent performance across all scenarios, particularly for CDL-C and D. However, a critical limitation of fsAD is its strict dependence on precise antenna array geometry, which is often unavailable in practical deployments. Similarly, while Compressed Sensing (CS) based approaches align with the structural sparsity of CDL models, their reliance on idealistic sparsity assumptions often proves infeasible in complex real-world environments.

#### D. Complexity Analysis

Fig. 3 illustrates the comprehensive trade-off among estimation accuracy (NMSE), the number of floating-point operations (FLOPs), and inference latency for the evaluated algorithms. Notably, FLOPs and latency are reported as the averaged time per sample, measured with a batch size of 100 on an NVIDIA A100 GPU. It is evident that the proposed RC-Flow achieves a superior balance between reconstruction fidelity and computational overhead. While the LMMSE and L-DAMP estimators offer extremely low latencies at the  $10^{-1}$  ms magnitude, they fail to provide competitive NMSE results. Conversely, the Score-based method and fsAD exhibit excessive latencies on the order of  $10^3$  ms, which exceeds the typical channel coherence time and renders them impractical for real-time deployments. Specifically, the Score-based approach involves



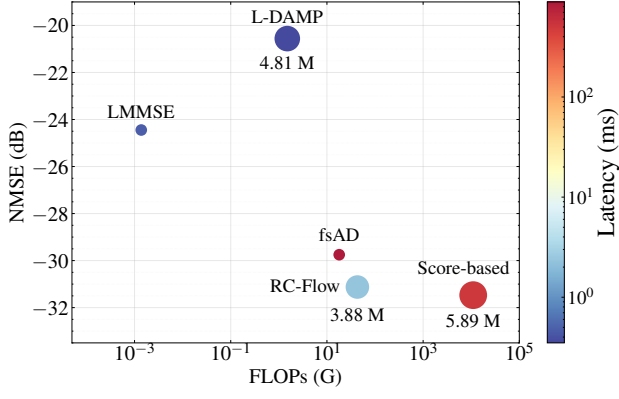


Fig. 3. Visualization of the trade-off between algorithm performance and computational cost. The evaluation is conducted with  $\alpha = 0.6$  at an SNR of 30 dB. The bubble size represents the scale of model parameters, and the color corresponds to the inference latency. The models were trained on CDL-C and tested on CDL-D.

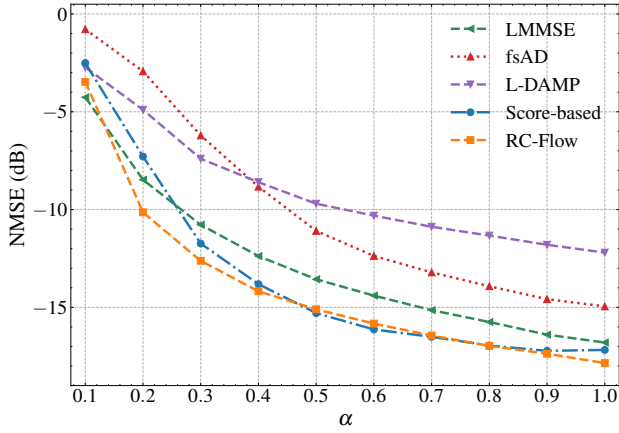


Fig. 4. Performance of different schemes trained on CDL-C channels and tested on CDL-D channels at an SNR of 10 dB with different pilot density.

a prohibitive FLOPs count at the  $10^4$  G level, necessitating substantial hardware resources. In contrast, the proposed RC-Flow delivers near-optimal NMSE with an inference latency at the  $10^0$  ms magnitude.

### E. System Scaling Analysis

*a) Performance Scaling with Pilot Overhead:* Fig. 4 illustrates the NMSE performance as a function of the pilot density  $\alpha \in \{0.1, 0.2, \dots, 1.0\}$ . All evaluated methods demonstrate consistent NMSE reduction as  $\alpha$  increases, with the proposed RC-Flow achieving a significant performance advantage. Specifically, the LMMSE estimator demonstrates robust performance across all pilot densities, showing particular resilience in the extremely low pilot density regime ( $\alpha = 0.1$ ). However, it still lags significantly behind both the RC-Flow and Score-based methods in the medium-to-high density regimes. In terms of L-DAMP, while it provides competitive results at very low pilot densities, its performance scales poorly with  $\alpha$ , resulting in marginal gains in the medium-to-high density regimes. Conversely, the fsAD method is more sensitive to pilot density, showing poor performance at low  $\alpha$  but improving faster as  $\alpha$  increases, though a substantial gap persists relative to the generative solvers. Notably, while RC-Flow and the Score-based method achieve similar precision

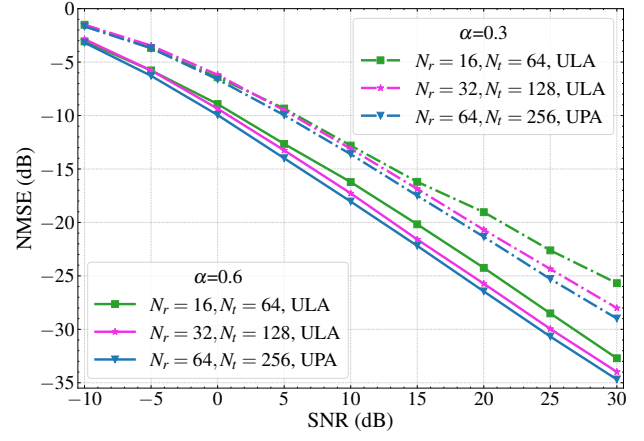


Fig. 5. Performance of the proposed schemes trained and tested on CDL-C channels with different MIMO size.

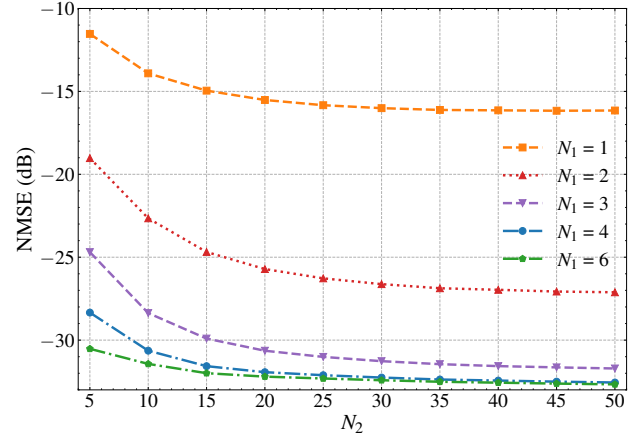


Fig. 6. Performance of the proposed scheme trained and tested on CDL-C channels at an SNR of 30 dB with different number of inner ( $N_2$ ) and outer loop iterations ( $N_1$ ), when  $\alpha = 0.6$ .

for  $\alpha > 0.4$ , RC-Flow demonstrates a noticeable advantage in the low-pilot regime, outperforming the Score-based baseline by 3 dB at  $\alpha = 0.2$ . These findings underscore the remarkable resilience of RC-Flow to limited pilot overhead.

*b) Scalability Across Massive MIMO Dimensions:* Fig. 5 illustrates the NMSE performance of RC-Flow across three distinct MIMO configurations with  $\alpha \in \{0.3, 0.6\}$ , while maintaining antenna spacing at half-wavelength. For each antenna dimension, we train a separate model individually. As observed, the estimation accuracy improves consistently as the MIMO dimensions scale up. This performance gain is primarily attributed to the superior angular resolution and the more pronounced structural sparsity inherent in larger antenna arrays. These characteristics allow the generative prior to provide more precise guidance within high-dimensional channel manifolds.

### F. Impact of Iteration Numbers

Fig. 6 illustrates the NMSE performance across varying outer and inner iteration numbers ( $N_1$  and  $N_2$ ) at an SNR of 30 dB with  $\alpha = 0.6$ . For a fixed  $N_1$ , the NMSE decreases significantly as  $N_2$  increases, typically plateauing beyond  $N_2 = 25$ . Notably, the marginal gains from increasing  $N_2$  are

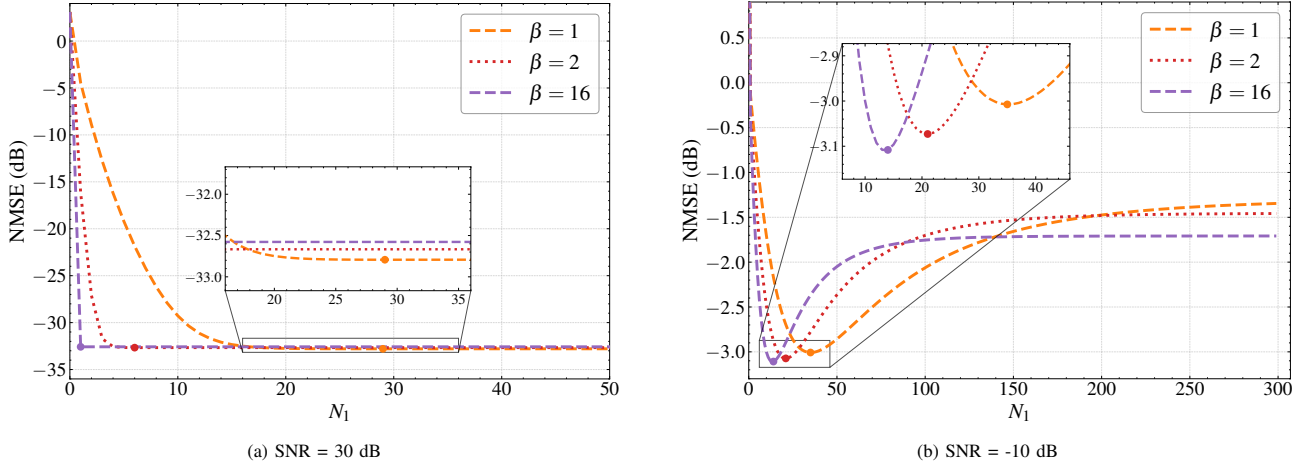


Fig. 7. The iterative process of the algorithm's dynamics with different  $\beta$  and  $\alpha = 0.6$ .

more pronounced at lower  $N_1$  values, whereas this influence diminishes as  $N_1$  becomes larger. For a fixed  $N_2$ , increasing  $N_1$  leads to significant improvements when  $N_1$  is small, with gains becoming marginal once  $N_1$  exceeds approximately 4 steps. These observations validate the efficacy of the proposed recursive anchor refinement mechanism. However, it is evident that both the outer processes and the inner fine-grained steps exhibit marginal effect, as the performance improvements gradually plateauing.

#### G. Analysis of Algorithm Dynamics

*a) Convergence Dynamics and  $\beta$  Tuning:* Figs. 7a and 7b illustrate the convergence dynamics of the proposed algorithm with  $\alpha = 0.6$  and  $\lambda = 2$  under different SNR regimes.

In the high SNR case, the NMSE exhibits a monotonic decreasing trend with respect to  $N_1$ , eventually stabilizing at a convergence floor. Increasing  $\beta$  significantly accelerates convergence. For instance, at  $\beta = 16$ , the algorithm achieves near-optimal performance within a single iteration. However, this acceleration incurs a marginal performance loss of approximately 0.1 dB compared to lower  $\beta$  settings. Conversely, at low SNR case, the dynamics follow a non-monotonic pattern: the NMSE initially descends to a minimum ‘sweet spot’, followed by a gradual ascent toward an asymptotic plateau. Notably, as  $\beta$  increases, the minimum point shifts toward the bottom-left, indicating simultaneous improvements in inference accuracy and speed. Furthermore, larger  $\beta$  values accelerate the transition to the plateau phase while achieving a lower converged NMSE.

A comparative analysis reveals that while the algorithm shares a common initialization dynamics, its asymptotic behavior is strictly governed by the signal quality. In the initial phase, both regimes exhibit a rapid descent in NMSE dominated by noise suppression, where the model effectively steers the estimate toward the inherently smooth posterior mean. However, subsequent behaviors diverge due to the perception-distortion trade-off [47]. At  $-10$  dB, the NMSE rebounds after reaching a minimum ‘sweet spot’ as the model proceeds to synthesize high-frequency structural details to satisfy manifold constraints. However, due to the ill-posed nature of the problem, these generated stochastic features suffer from spatial

misalignment, effectively prioritizing perceptual consistency at the expense of pixel-wise fidelity. Conversely, as the SNR increases to 30 dB, reliable measurements ensure that the synthesized fine-grained textures remain faithful to the true channel state. Consequently, the convergence region aligns with the sweet spot, avoiding the distortion penalty associated with hallucinatory features and resulting in the strictly monotonic trajectory where no early stopping is required.

*b) Sensitivity analysis of  $\lambda$ - $\beta$  configurations:* Fig. 8 illustrates a comprehensive sensitivity analysis of RC-Flow regarding hyperparameters  $\lambda$  and  $\beta$  at  $\alpha = 0.6$ . The heatmaps (a) – (c) represent the NMSE at the sweet spot, while the line graphs (d) – (f) depict the outer iteration numbers required to reach this point within the maximum limit of  $N_1 = 100$ .

Analysis of the heatmaps and line graphs reveal a distinct correspondence between reconstruction fidelity and convergence efficiency. Regarding NMSE performance, optimal results are consistently achieved in the moderate  $\lambda$  regime, while larger  $\lambda$  values significantly decelerate the convergence process. This occurs because an excessively high  $\lambda$  causes the time variable  $t$  to decrease too rapidly, diminishing the flow-predicted velocity and resulting in smaller per-step updates. Furthermore, while increasing  $\beta$  generally accelerates convergence, enabling RC-Flow to reach the sweet spot in a single  $N_1$  step under medium-to-high SNR scenarios, it induces a slight performance degradation of approximately 1 dB in medium SNR regimes. In the low-SNR regime, at least  $N_1 = 6$  iterations are required under an optimal  $\lambda$ - $\beta$  configurations. This suggests that higher  $\beta$  values prioritize convergence speed, potentially at the expense of absolute precision in intermediate noise conditions.

Finally, we highlight that certain  $\lambda$ - $\beta$  configurations, particularly in the upper-left regions of the heatmaps. The sharp performance degradation in these regions does not necessarily imply that such  $\lambda$ - $\beta$  combinations result in poor potential performance, but rather that the trajectory has not yet reached the sweet spot within the maximum  $N_1$  limitation. This underscores the significance of balanced scheduling to optimize both reconstruction fidelity and computational efficiency.

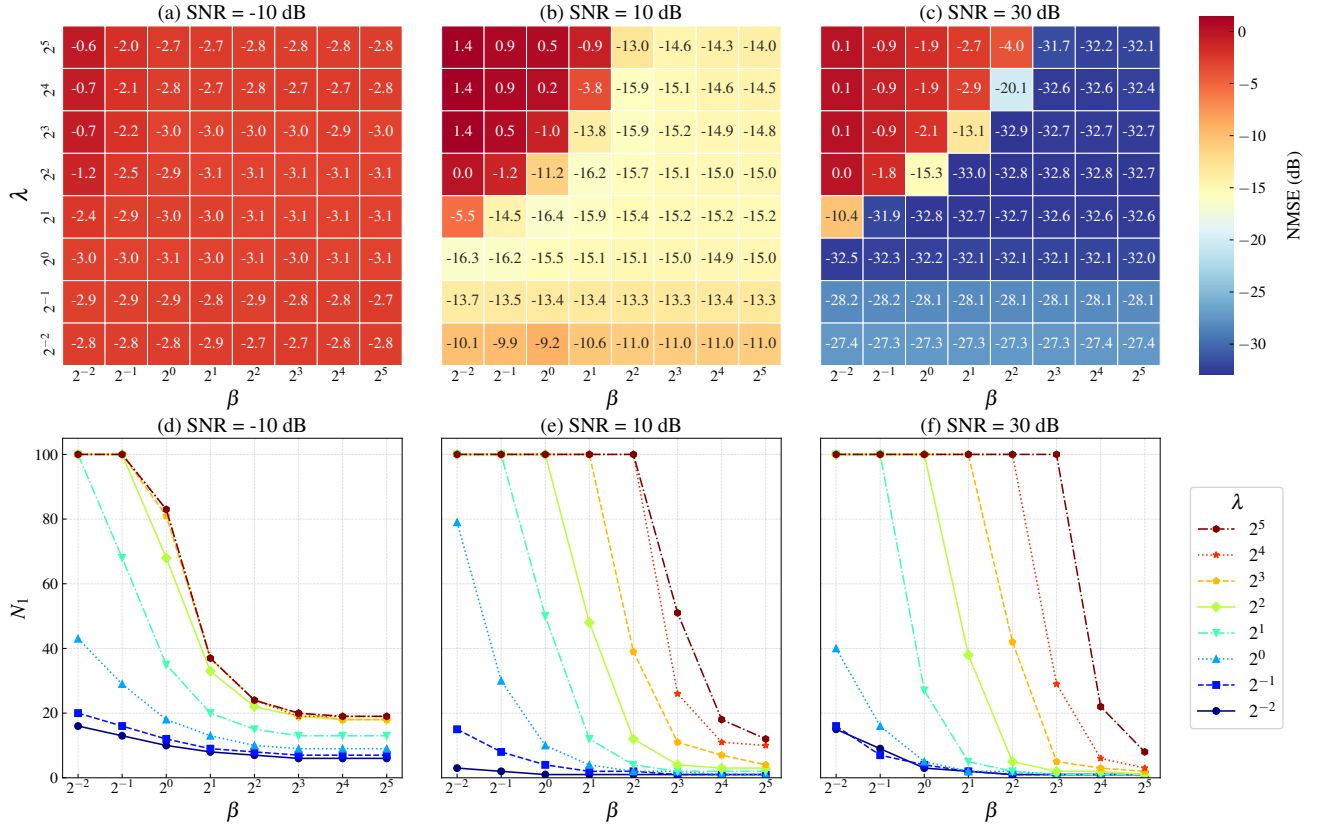


Fig. 8. Sensitivity analysis of hyperparameters  $\lambda$  (Flow Schedule) and  $\beta$  (Anchor Schedule) across different SNR regimes. The heatmap displays the NMSE (dB), and the line graph displays the least number of outer iterations required to reach the sweet spot as shown in Fig. 7.

## V. CONCLUSION

In this paper, we presented RC-Flow, a unified generative framework for high-dimensional MIMO channel estimation. By reformulating the inference process as a closed-loop fixed-point iteration, our method addresses the fundamental limitations of open-loop generative samplers in ill-posed inverse problems. The core novelty lies in the serial restart mechanism combined with anchored trajectory rectification, which effectively suppresses error propagation, even under extreme noise conditions. This design allows RC-Flow to achieve SOTA accuracy across a wide range of SNRs and pilot density, reducing inference latency by orders of magnitude compared to score-based approach. We believe this framework establishes a new paradigm for solving constrained inverse problems, paving the way for low-latency, high-fidelity generative inference in 6G communications.

## APPENDIX A PROOF OF THEOREM 1

To invoke Brouwer's Fixed-Point Theorem, we must identify a compact convex ball  $\mathcal{K}_{R^*}$  such that  $\mathcal{T}(\mathcal{K}_{R^*}) \subseteq \mathcal{K}_{R^*}$ .

### A. Continuity

The neural network  $\mathcal{D}$ , the linear matrix operations in the projection  $\mathcal{P}$ , and the convex combination with the anchor are all continuous mappings. Since the matrix inverse  $(\mathbf{M} + w^{-1}\mathbf{I})^{-1}$  exists and is continuous for  $w^{-1} > 0$ , the composite operator  $\mathcal{T}$  is continuous on  $\mathbb{C}^{N_r \times N_t}$ .

### B. Self-Mapping on a Compact Set

We establish a global upper bound on the projected states, independent of the iteration index. Based on Assumption 1, the output is bounded by  $\|\mathcal{D}(\mathbf{H})\|_F \leq B_{\text{flow}}$ . Consider the projection step at any inner index  $i$ :

$$\mathbf{H}_{\text{proj}}^{(i)} = (\mathbf{R} + w^{-1}\mathcal{D}(\mathbf{H}^{(k,i)}))(\mathbf{M} + w^{-1}\mathbf{I})^{-1}. \quad (29)$$

Applying the norm inequality and the spectral bound  $\|w^{-1}(\mathbf{M} + w^{-1}\mathbf{I})^{-1}\|_2 \leq 1$ :

$$\begin{aligned} \|\mathbf{H}_{\text{proj}}^{(i)}\|_F &\leq \|\mathbf{R}(\mathbf{M} + w^{-1}\mathbf{I})^{-1}\|_F + \|\mathcal{D}(\cdot)\|_F \cdot 1 \\ &\leq C_{\text{obs}} + B_{\text{flow}} \triangleq R_{\text{limit}}. \end{aligned} \quad (30)$$

Crucially,  $R_{\text{limit}}$  is a constant determined solely by the observations and the physical prior manifold. It does not depend on the norm of the intermediate state  $\mathbf{H}^{(k,i)}$  or the step index  $i$ .

Define the invariant ball  $\mathcal{K}_{R_{\text{limit}}} \triangleq \{\mathbf{H} \mid \|\mathbf{H}\|_F \leq R_{\text{limit}}\}$ , which is inherently a convex set. Assume the initialization satisfies  $\mathbf{H}^{(k,0)} \in \mathcal{K}_{R_{\text{limit}}}$ , implying the anchor  $\epsilon \in \mathcal{K}_{R_{\text{limit}}}$ . Since we have established that any projected state is bounded by  $R_{\text{limit}}$ , we also have  $\mathbf{H}_{\text{proj}}^{(i)} \in \mathcal{K}_{R_{\text{limit}}}$ . The update rule in (17) constitutes a convex combination. By the definition of convexity, the resulting state  $\mathbf{H}^{(k,i+1)}$  must remain within  $\mathcal{K}_{R_{\text{limit}}}$  for any  $t'_i \in [0, 1)$ . By induction, the final output satisfies  $\mathbf{H}^{(k,N_2)} \in \mathcal{K}_{R_{\text{limit}}}$ .

Therefore, we have proved that if the input  $\mathbf{H}^{(k,0)} \in \mathcal{K}_{R^*}$ , then the output of the composite inner loop  $\mathcal{T}(\mathbf{H}^{(k,0)}) \in \mathcal{K}_{R^*}$ . Thus,  $\mathcal{T}$  is a self-mapping on the compact convex set  $\mathcal{K}_{R^*}$ .

By Brouwer's Fixed-Point Theorem, there exists at least one fixed point  $\mathbf{H}^* \in \mathcal{K}_{R^*}$ .

## APPENDIX B PROOF OF THEOREM 2

In this appendix, we provide a rigorous derivation of the global asymptotic stability in a generalized metric space for the RC-Flow algorithm. The objective is to bound the spectral radius of the total Jacobian matrix  $\mathbf{J}_{\mathcal{T}} \in \mathbb{C}^{N \times N}$  (where  $N = N_t N_r$ ), which governs the sensitivity of the output of the  $k$ -th outer iteration,  $\mathbf{H}^{(k+1,0)}$ , with respect to its input,  $\mathbf{H}^{(k,0)}$ .

### A. Stability Criterion via Induced Metric Space

We invoke Assumption 2, which postulates that the operator  $\mathbf{T}_i \triangleq \mathbf{J}_{\mathcal{P},i} \mathbf{J}_{\mathcal{D},i}$  satisfies the spectral contraction condition  $\rho(\mathbf{T}_i) \leq \gamma < 1$ . According to the Householder theorem, for any matrix  $\mathbf{A}$  with  $\rho(\mathbf{A}) < 1$  and any given  $\eta > 0$ , there exists an induced vector norm  $\|\cdot\|_*$  such that:

$$\|\mathbf{A}\|_* \leq \rho(\mathbf{A}) + \eta. \quad (31)$$

However, since the operator  $\mathbf{T}_i$  is time-varying, strictly establishing stability requires a common metric. Given the smooth evolution of the parameters, we analytically extend Assumption 2 to postulate the existence of a common induced norm  $\|\cdot\|$  for the sequence  $\{\mathbf{T}_i\}_{i=0}^{N_2-1}$ , such that:

$$\|\mathbf{T}_i\|_* \leq \gamma' < 1, \quad \forall i. \quad (32)$$

Consequently, our proof strategy is to establish the contraction of the total Jacobian  $\mathbf{J}_{\mathcal{T}}$  within this generalized metric space.

### B. Differential Dynamics of the Inner Loop

Consider the  $k$ -th outer loop iteration. Let the input state be  $\mathbf{H}^{(k,0)}$ . According to Algorithm 1, the anchor point is reset to the previous estimate, i.e.,  $\epsilon = \mathbf{H}^{(k,0)}$ . The inner loop generates a sequence of states  $\{\mathbf{H}^{(k,i)}\}_{i=0}^{N_2}$ , where the update rule at the  $i$ -th inner step is given by:

$$\mathbf{H}^{(k,i+1)} = t'_i \mathbf{H}^{(k,0)} + (1 - t'_i) \mathcal{P}(\mathcal{D}(\mathbf{H}^{(k,i)})), \quad (33)$$

We define the cumulative sensitivity matrix at inner step  $i$  as  $\mathbf{J}^{(i)} \triangleq \frac{\partial \text{vec}(\mathbf{H}^{(k,i)})}{\partial \text{vec}(\mathbf{H}^{(k,0)})}$ . By applying the chain rule to the recurrence relation and using the definition of the composite Jacobian  $\mathbf{T}_i$ , we obtain:

$$\mathbf{J}^{(i+1)} = t'_i \mathbf{I} + (1 - t'_i) \mathbf{T}_i \mathbf{J}^{(i)}. \quad (34)$$

where  $\mathbf{I}$  is the identity matrix resulting from differentiating the anchor term  $t'_i \mathbf{H}^{(k,0)}$ .

### C. Recursive Expansion and Closed-Form Solution

The base case for the recurrence is  $\mathbf{J}^{(0)} = \mathbf{I}$ . To reveal the structure of the total Jacobian, let us define the path transition operator  $\Phi_{m,j}$ , which represents the accumulated decay and rotation from step  $j$  to  $m$ :

$$\Phi_{m,j} \triangleq \begin{cases} \prod_{n=j}^{m-1} (1 - t'_n) \mathbf{T}_n, & \text{if } m > j, \\ \mathbf{I}, & \text{if } m = j. \end{cases} \quad (35)$$

By induction, the total Jacobian  $\mathbf{J}_{\mathcal{T}} \triangleq \mathbf{J}^{(N_2)}$  at the end of the inner loop can be expressed as a weighted sum of historical transition paths:

$$\mathbf{J}_{\mathcal{T}} = \Phi_{N_2,0} + \sum_{i=0}^{N_2-1} t'_i \Phi_{N_2,i+1}. \quad (36)$$

### D. Strict Contraction Analysis via Partition of Unity

We now apply the induced norm  $\|\cdot\|_*$  defined in (32) to the closed-form expansion (36), we obtain:

$$\|\mathbf{J}_{\mathcal{T}}\|_* \leq \|\Phi_{N_2,0}\|_* + \sum_{i=0}^{N_2-1} t'_i \|\Phi_{N_2,i+1}\|_*. \quad (37)$$

The norm of the transition operator is bounded by the product of individual operator norms. Using  $\|\mathbf{T}_n\|_* \leq \gamma' < 1$ , we have:

$$\|\Phi_{N_2,j}\|_* \leq \prod_{n=j}^{N_2-1} (1 - t'_n) \|\mathbf{T}_n\|_* \leq (\gamma')^{N_2-j} \prod_{n=j}^{N_2-1} (1 - t'_n). \quad (38)$$

For brevity, we define the cumulative scalar decay factor  $P_j$ :

$$P_j \triangleq \prod_{n=j}^{N_2-1} (1 - t'_n), \quad \text{with } P_{N_2} \triangleq 1. \quad (39)$$

Substituting this bound into the inequality (37), we arrive at:

$$\|\mathbf{J}_{\mathcal{T}}\|_* \leq P_0 (\gamma')^{N_2} + \sum_{i=0}^{N_2-1} t'_i P_{i+1} (\gamma')^{N_2-(i+1)}. \quad (40)$$

To prove strict contraction, we utilize the algebraic property of the weighting coefficients, yielding:

$$P_0 + \sum_{i=0}^{N_2-1} t'_i P_{i+1} = P_0 + \sum_{i=0}^{N_2-1} (P_{i+1} - P_i) = P_{N_2} = 1. \quad (41)$$

Therefore, the upper bound in (40) represents a convex combination of the terms  $(\gamma')^k$ .

Since  $\gamma' < 1$  and  $N_2 \geq 1$ , the factor  $(\gamma')^k$  is strictly less than 1 for all  $k \geq 1$ . Specifically, the exponents in (40) are  $N_2, N_2 - 1, \dots, 0$ . Unless  $t'_{N_2-1} = 1$ , the sum contains terms scaled by  $\gamma' < 1$ . Thus, the strict inequality holds:

$$\|\mathbf{J}_{\mathcal{T}}\|_* < P_0 \cdot 1 + \sum_{i=0}^{N_2-1} t'_i P_{i+1} \cdot 1 = 1. \quad (42)$$

Since  $\|\mathbf{J}_{\mathcal{T}}\|_* < 1$ , the mapping is a strict contraction in the metric space  $(\mathbb{C}^N, \|\cdot\|_*)$ . Therefore, the proof is complete.

## REFERENCES

- [1] X. You, C.-X. Wang, J. Huang *et al.*, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China Inf. Sci.*, vol. 64, no. 1, p. 110301, 2021.
- [2] C.-X. Wang, X. You, X. Gao *et al.*, "On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds," *IEEE Commun. Surv. Tutor.*, vol. 25, no. 2, pp. 905–974, 2023.
- [3] E. G. Larsson, O. Edfors, F. Tufvesson *et al.*, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, 2014.

- [4] L. Lu, G. Y. Li, A. L. Swindlehurst *et al.*, “An Overview of Massive MIMO: Benefits and Challenges,” *IEEE J. Sel. Top. Signal Process.*, vol. 8, no. 5, pp. 742–758, 2014.
- [5] J. Tian, Y. Han, S. Jin *et al.*, “Analytical channel modeling: From MIMO to extra large-scale MIMO,” *Chin. J. of Electron.*, vol. 34, no. 1, pp. 1–15, 2025.
- [6] K. Venugopal, A. Alkhateeb, N. González Prelcic *et al.*, “Channel Estimation for Hybrid Architecture-Based Wideband Millimeter Wave Systems,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1996–2009, 2017.
- [7] A. Alkhateeb, O. El Ayach, G. Leus *et al.*, “Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems,” *IEEE J. Sel. Top. Signal Process.*, vol. 8, no. 5, pp. 831–846, 2014.
- [8] P. Schniter and A. Sayeed, “Channel estimation and precoder design for millimeter-wave communications: The sparse way,” in *Asilomar Conf. Signals, Syst. Comput.*, 2014, pp. 273–277.
- [9] D. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [10] E. Nayeibi and B. D. Rao, “Semi-blind Channel Estimation for Multiuser Massive MIMO Systems,” *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 540–553, 2018.
- [11] B. N. Bhaskar, G. Tang, and B. Recht, “Atomic Norm Denoising With Applications to Line Spectral Estimation,” *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5987–5999, 2013.
- [12] H. He, C.-K. Wen, S. Jin *et al.*, “Deep Learning-Based Channel Estimation for BeamSpace mmWave Massive MIMO Systems,” *IEEE Wirel. Commun. Lett.*, vol. 7, no. 5, pp. 852–855, 2018.
- [13] M. Soltani, V. Pourahmadi, A. Mirzaei *et al.*, “Deep Learning-Based Channel Estimation,” *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, 2019.
- [14] C. Metzler, A. Mousavi, and R. Baraniuk, “Learned D-AMP: Principled neural network based compressive image recovery,” in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [15] K. Zhang, W. Zuo, Y. Chen *et al.*, “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising,” *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [16] D. Fan, R. Meng, X. Xu *et al.*, “Generative diffusion models for wireless networks: Fundamental, architecture, and state-of-the-art,” *arXiv preprint arXiv:2507.16733*, 2025.
- [17] F. Khoramnejad and E. Hossain, “Generative AI for the Optimization of Next-Generation Wireless Networks: Basics, State-of-the-Art, and Open Challenges,” *IEEE Commun. Surv. Tutor.*, vol. 27, no. 6, pp. 3483–3525, 2025.
- [18] A. Shahid, A. Kliks, A. Al-Tahmeesschi *et al.*, “Large-scale AI in telecom: Charting the roadmap for innovation, scalability, and enhanced digital experiences,” *arXiv preprint arXiv:2503.04184*, 2025.
- [19] F. Zhu, X. Wang, S. Jiang *et al.*, “Wireless large AI model: Shaping the AI-native future of 6G and beyond,” *arXiv preprint arXiv:2504.14653*, 2025.
- [20] J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models,” in *Int. Conf. Learn. Represent.*, H. Larochelle, M. Ranzato, R. Hadsell *et al.*, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.
- [21] Y. Song, J. Sohl-Dickstein, D. P. Kingma *et al.*, “Score-Based Generative Modeling through Stochastic Differential Equations,” in *Int. Conf. Learn. Represent.*, 2021.
- [22] T. Lee, J. Park, H. Kim *et al.*, “Generating high dimensional user-specific wireless channels using diffusion models,” *IEEE Trans. Wirel. Commun.*, pp. 1–1, 2025.
- [23] T. Wu, Z. Chen, D. He *et al.*, “CDDM: Channel Denoising Diffusion Models for Wireless Semantic Communications,” *IEEE Trans. Wirel. Commun.*, vol. 23, no. 9, pp. 11 168–11 183, 2024.
- [24] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu *et al.*, “Flow Matching for Generative Modeling,” in *Int. Conf. Learn. Represent.*, 2023.
- [25] G. Papamakarios, E. Nalisnick, D. J. Rezende *et al.*, “Normalizing Flows for Probabilistic Modeling and Inference,” *J. Mach. Learn. Res.*, vol. 22, no. 57, pp. 1–64, 2021.
- [26] I. Kobyzev, S. J. Prince, and M. A. Brubaker, “Normalizing Flows: An Introduction and Review of Current Methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3964–3979, 2021.
- [27] X. Liu, C. Gong, and Q. Liu, “Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow,” in *Int. Conf. Learn. Represent.*, 2023.
- [28] E. F. Montesuma, F. M. N. Mboula, and A. Souloumiac, “Recent Advances in Optimal Transport for Machine Learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 2, pp. 1161–1180, 2025.
- [29] M. Pourya, B. E. Rawas, and M. Unser, “FLOWER: A Flow-Matching Solver for Inverse Problems,” *arXiv preprint arXiv:2509.26287*, 2025.
- [30] S. T. Martin, A. Gagneux, P. Hagemann *et al.*, “PnP-Flow: Plug-and-Play Image Restoration with Flow Matching,” in *Int. Conf. Learn. Represent.*, 2025.
- [31] W. Liu, N. Ma, J. Chen *et al.*, “Flow matching-based generative models for MIMO channel estimation,” *arXiv preprint arXiv:2511.10941*, 2025.
- [32] B. Fesl, M. Baur, F. Strasser *et al.*, “Diffusion-Based Generative Prior for Low-Complexity MIMO Channel Estimation,” *IEEE Wirel. Commun. Lett.*, vol. 13, no. 12, pp. 3493–3497, 2024.
- [33] Z. Jiang, F. Zhu, S. Jiang *et al.*, “One-Step Generative Channel Estimation via Average Velocity Field,” in *IEEE Wirel. Commun. Netw. Conf. (WCNC)*, 2026.
- [34] M. Arvinte and J. I. Tamir, “MIMO Channel Estimation Using Score-Based Generative Models,” *IEEE Trans. Wirel. Commun.*, vol. 22, no. 6, pp. 3698–3713, 2023.
- [35] X. Zhou, L. Liang, J. Zhang *et al.*, “Generative Diffusion Models for High Dimensional Channel Estimation,” *IEEE Trans. Wirel. Commun.*, vol. 24, no. 7, pp. 5840–5854, 2025.
- [36] Z. Chen, H. Shin, and A. Nallanathan, “Generative Diffusion Model-Based Variational Inference for MIMO Channel Estimation,” *IEEE Trans. Commun.*, vol. 73, no. 10, pp. 9254–9269, 2025.
- [37] Z. Diao, X. Zhou, L. Liang *et al.*, “Robust MIMO Channel Estimation Using Energy-Based Generative Diffusion Models,” *IEEE Wirel. Commun. Lett.*, vol. 15, pp. 820–824, 2026.
- [38] B. Efron, “Tweedie’s formula and selection bias,” *J. Am. Stat. Assoc.*, vol. 106, no. 496, pp. 1602–1614, 2011.
- [39] K. Kim and J. C. Ye, “Noise2score: tweedie’s approach to self-supervised image denoising without clean images,” *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 864–874, 2021.
- [40] S. Bai, J. Z. Kolter, and V. Koltun, “Deep equilibrium models,” *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
- [41] Z. Jiang and F. Zhu, “RC-Flow,” <https://github.com/jzzh123/RC-Flow>, 2025.
- [42] E. Tamir, N. Laabid, M. Heinonen *et al.*, “Conditional Flow Matching for Time Series Modelling,” in *ICML Workshop Struct. Prob. Infer. Gen. Model.*, 2024.
- [43] S. H. Chan, X. Wang, and O. A. Elgandy, “Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications,” *IEEE Trans. Comput. Imaging*, vol. 3, no. 1, pp. 84–98, 2017.
- [44] R. B. Kellogg, T.-Y. Li, and J. Yorke, “A Constructive Proof of the Brouwer Fixed-Point Theorem and Computational Results,” *SIAM J. Numer. Anal.*, vol. 13, no. 4, pp. 473–483, 1976.
- [45] 3GPP, “Study on channel model for frequencies from 0.5 to 100 GHz,” 2020, 3GPP TR 38.901 version 16.1.0 Release 16.
- [46] K. He, X. Zhang, S. Ren *et al.*, “Deep residual learning for image recognition,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
- [47] Y. Blau and T. Michaeli, “The Perception-Distortion Tradeoff,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 6228–6237.