

Semi-Blind MIMO Channel Estimation Using Diffusion Models

Xinyuan Wang and Krishna Narayanan

Professor, Department of Electrical and Computer Engineering
Co-Director, Texas A&M Initiative for Connected Intelligence

March 03, 2026



What is (Deep) Generative Modeling?

- ▶ $X \in \mathbb{R}^d$ is a random vector with a complicated density function $q_X(x)$
- ▶ For example, X might represent an image (flattened into a vector)
- ▶ *We are interested in $X = H$ - MIMO channel*
- ▶ We don't know $q_X(x)$ - we have a dataset with samples from $q_X(x)$ given by

$$\mathcal{D} = \{x^1, x^2, \dots, x^M\}$$

What is (Deep) Generative Modeling?

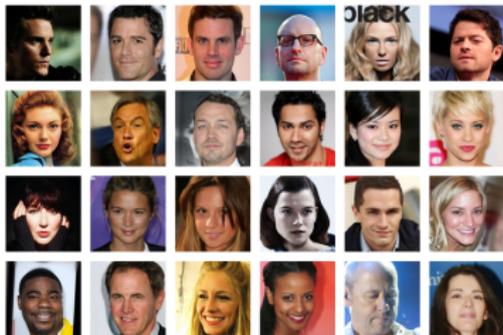
- ▶ $X \in \mathbb{R}^d$ is a random vector with a complicated density function $q_X(x)$
- ▶ For example, X might represent an image (flattened into a vector)
- ▶ *We are interested in $X = H$ - MIMO channel*
- ▶ We don't know $q_X(x)$ - we have a dataset with samples from $q_X(x)$ given by

$$\mathcal{D} = \{x^1, x^2, \dots, x^M\}$$

- ▶ What can we do with generative modeling?
 - Generate more samples from $q_X(x)$
 - Solve inverse problems - given $y = f(x) + n$, estimate x
 - At low SNR, we have to rely on the prior for x - often assume stylized models
 - Sometimes side information about x may be available

Generating New Samples

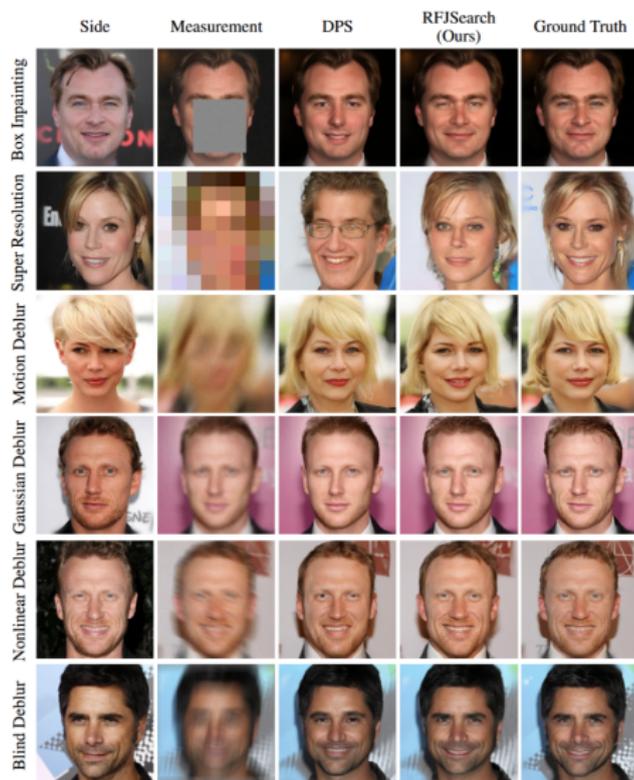
CelebA-HQ samples



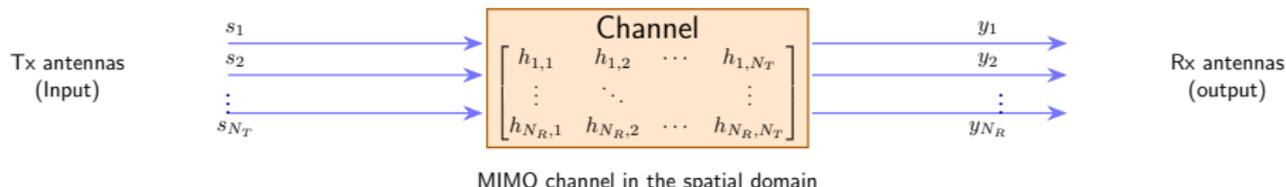
DDPM generated samples



Inverse Problems



Semi-Blind Channel Estimation Problem



$$\text{Pilots: } \mathbf{Y}_p = \mathbf{H}\mathbf{S}_p + \mathbf{W}_p, \quad (1)$$

$$\text{Data: } \mathbf{Y}_d = \mathbf{H}\mathbf{S}_d + \mathbf{W}_d. \quad (2)$$

- ▶ Channel in spatial domain - $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$
- ▶ User transmits $[\mathbf{S}_p \ \mathbf{S}_d]$
- ▶ Pilot matrix - $\mathbf{S}_p \in \mathbb{C}^{N_T \times N_p}$, $\mathbf{Y}_p \in \mathbb{C}^{N_R \times N_p}$
- ▶ Data matrix - $\mathbf{S}_d \in \mathbb{C}^{N_T \times N_d}$, $\mathbf{Y}_d \in \mathbb{C}^{N_R \times N_d}$

Problem Definition

- ▶ Given $\mathbf{Y}_p, \mathbf{S}_p, \mathbf{Y}_d$ estimate \mathbf{H} and detect \mathbf{S}_d
- ▶ **Prior distribution of \mathbf{H} can be complicated**

Problem Setup

$$\text{Pilots: } \mathbf{Y}_p = \mathbf{H}\mathbf{S}_p + \mathbf{W}_p, \quad (3)$$

$$\text{Data: } \mathbf{Y}_d = \mathbf{H}\mathbf{S}_d + \mathbf{W}_d. \quad (4)$$

Desired Properties

- ▶ A data-driven solution that does not assume stylized models - Diffusion models
- ▶ A robust solution that uses \mathbf{Y}_d well for any N_d
- ▶ We don't want to train for any specific N_d
- ▶ Algorithms that preserve the matrix structure of \mathbf{H} rather than $\text{vec}(\mathbf{H})$

Spatial, Angular Domains, and Gram Matrices

- ▶ Let $\tilde{\mathbf{H}}$ denote the 2-D DFT of \mathbf{H} where Φ_N is the 1-D DFT matrix

$$\tilde{\mathbf{H}} = \Phi_{N_R} \mathbf{H} \Phi_{N_T}^T, \quad (5)$$

- ▶ Variables with tilde refer to matrices in the angular domain

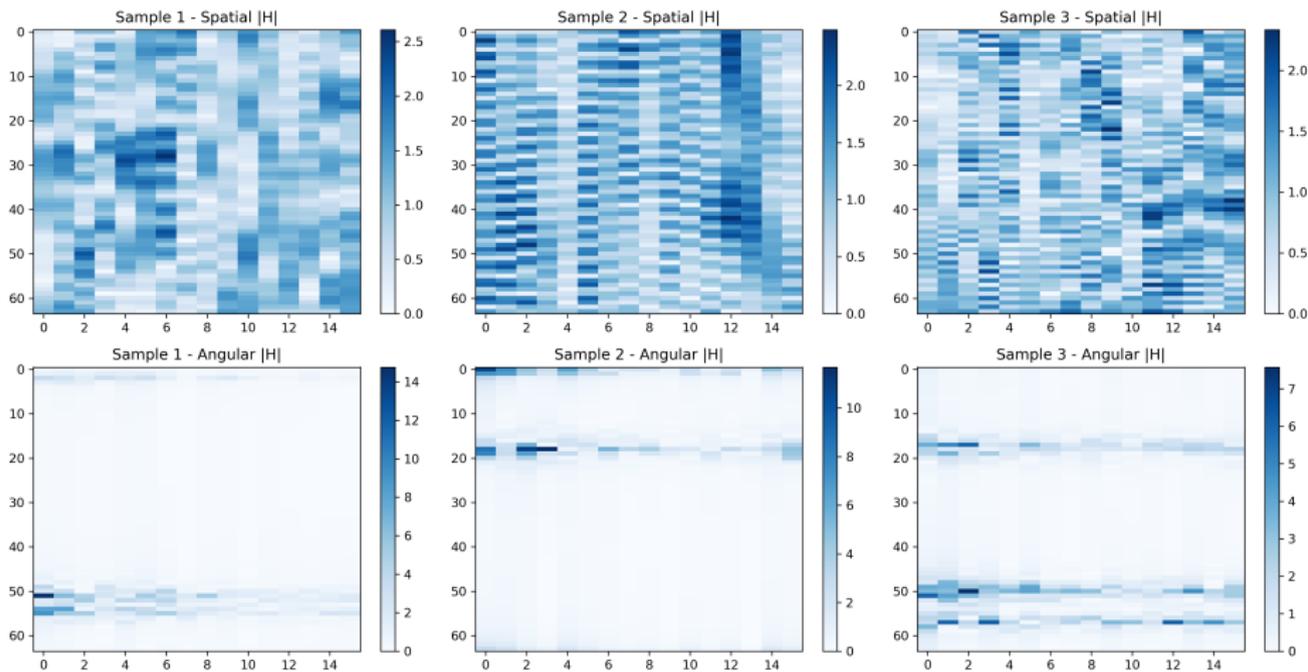
- ▶ Let $\mathbf{R} := \mathbf{H}\mathbf{H}^H$ and $\tilde{\mathbf{R}} := \tilde{\mathbf{H}}\tilde{\mathbf{H}}^H$ denote Gram matrices of rows of \mathbf{H} and $\tilde{\mathbf{H}}$

$$\tilde{\mathbf{R}} = \tilde{\mathbf{H}}\tilde{\mathbf{H}}^H = \Phi_{N_R} \mathbf{H} \Phi_{N_T}^T (\Phi_{N_T}^T)^H \mathbf{H}^H \Phi_{N_R}^H \quad (6)$$

$$= \Phi_{N_R} \mathbf{R} \Phi_{N_R}^H. \quad (7)$$

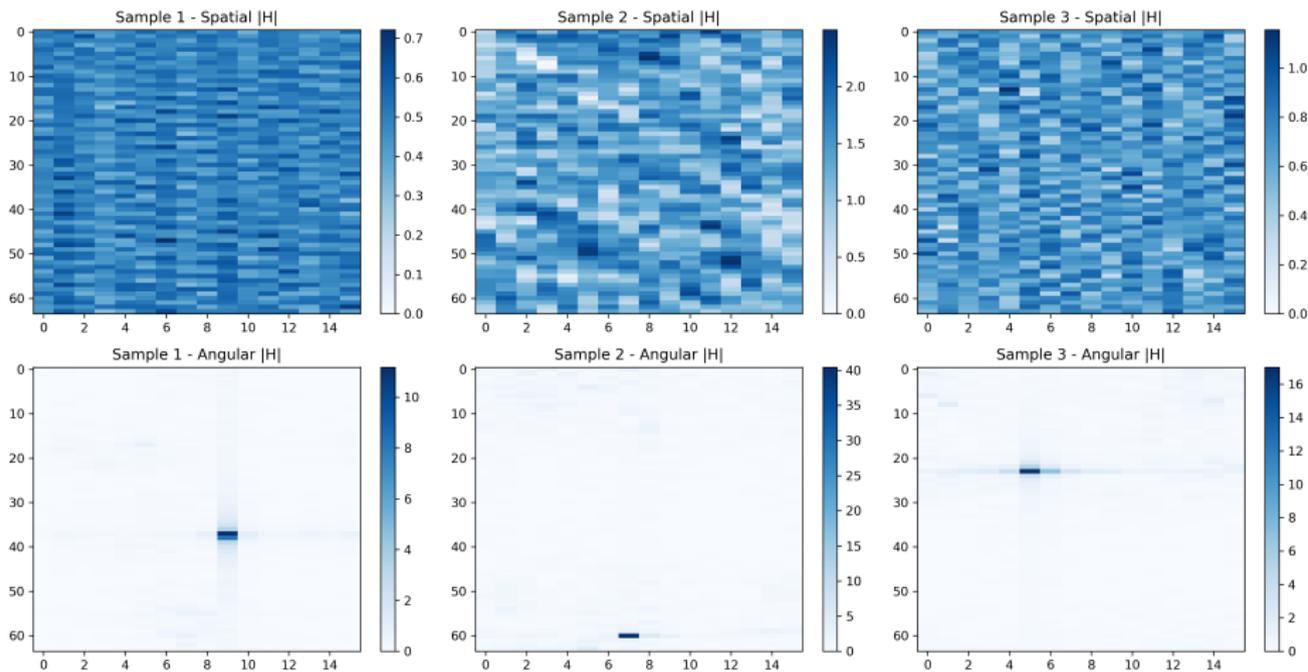
Sample MIMO Channel Realizations - 3GPP Dataset

3GPP Magnitude $|H|$: Spatial (top) vs Angular (bottom)

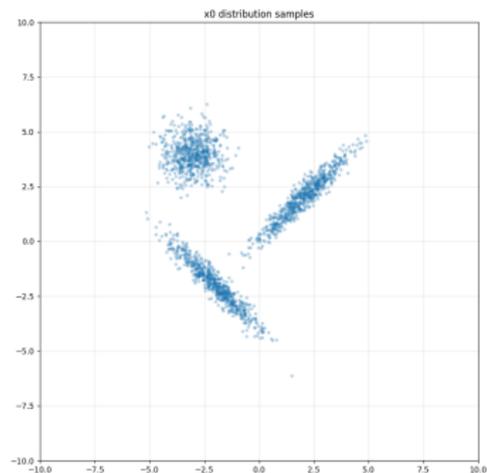
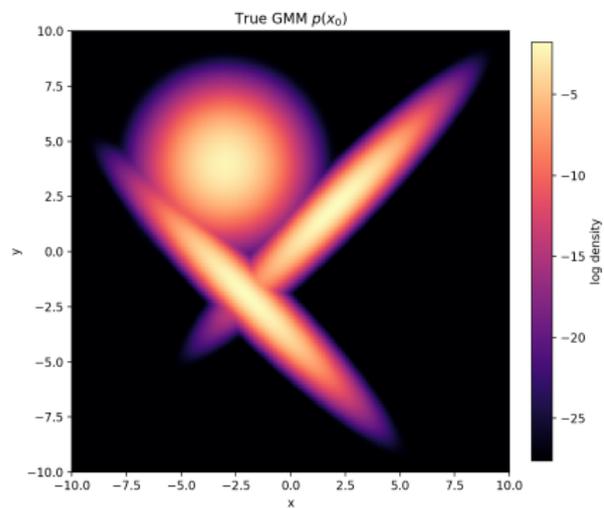


Sample MIMO Channel Realizations - Quadriga Dataset

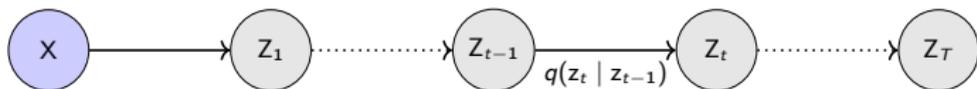
Quadriga Magnitude $|H|$: Spatial (top) vs Angular (bottom)



Mixture Gaussian Model



Forward Model



$$Z_1 = \sqrt{1 - \beta_1} X + \sqrt{\beta_1} \eta_1, \quad \eta_1 \sim \mathcal{N}(0, I) \quad (8)$$

$$Z_2 = \sqrt{1 - \beta_2} Z_1 + \sqrt{\beta_2} \eta_2 \quad (9)$$

$$= \sqrt{(1 - \beta_1)(1 - \beta_2)} X + \underbrace{\sqrt{(1 - \beta_2)\beta_1} \eta_1 + \sqrt{\beta_2} \eta_2}_{\eta_{eq}} \quad (10)$$

We can verify by induction that

$$Z_t = \sqrt{1 - \beta_t} Z_{t-1} + \sqrt{\beta_t} \eta_t \quad (11)$$

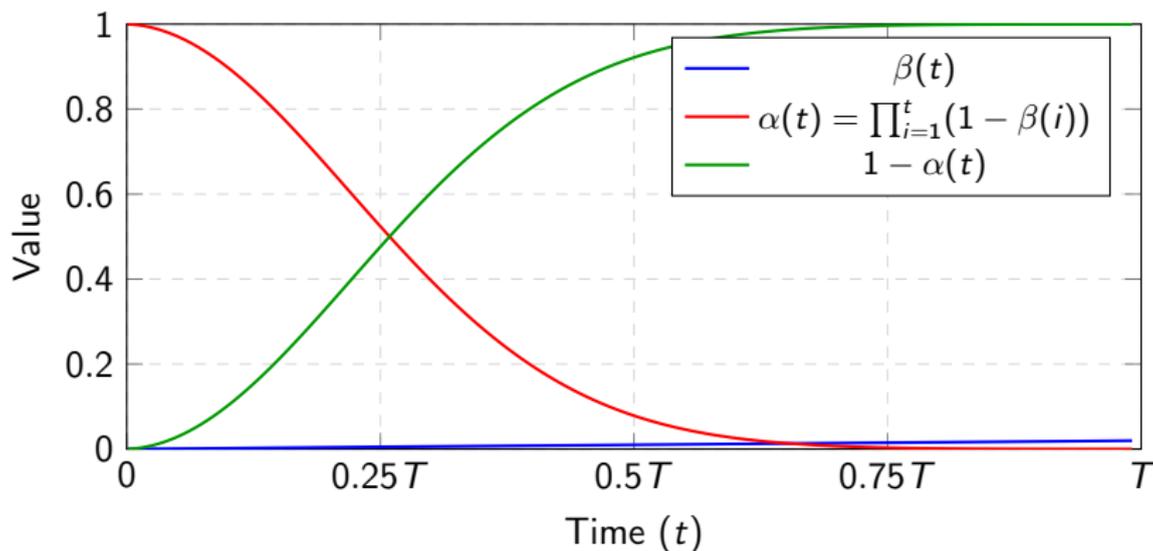
$$Z_t = \sqrt{\alpha_t} X + \sqrt{1 - \alpha_t} \epsilon_t \quad (12)$$

where $\alpha_t = \prod_{\tau=1}^t (1 - \beta_\tau)$ and η_t and ϵ_t are $\sim \mathcal{N}(0, I)$.

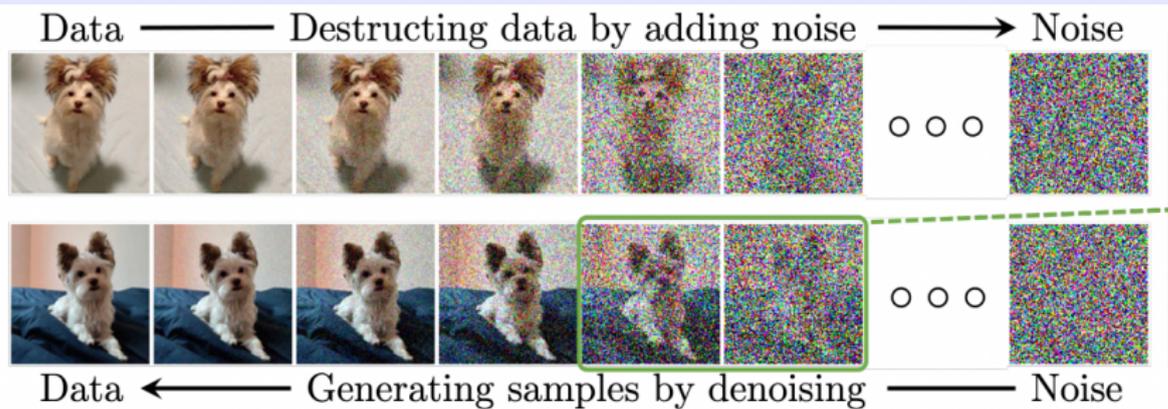
Noise Schedule - Plot of $\beta_t, \alpha_t, 1 - \alpha_t$ vs t

$$Z_t = \sqrt{1 - \beta_t} Z_{t-1} + \sqrt{\beta_t} \eta_t \quad (13)$$

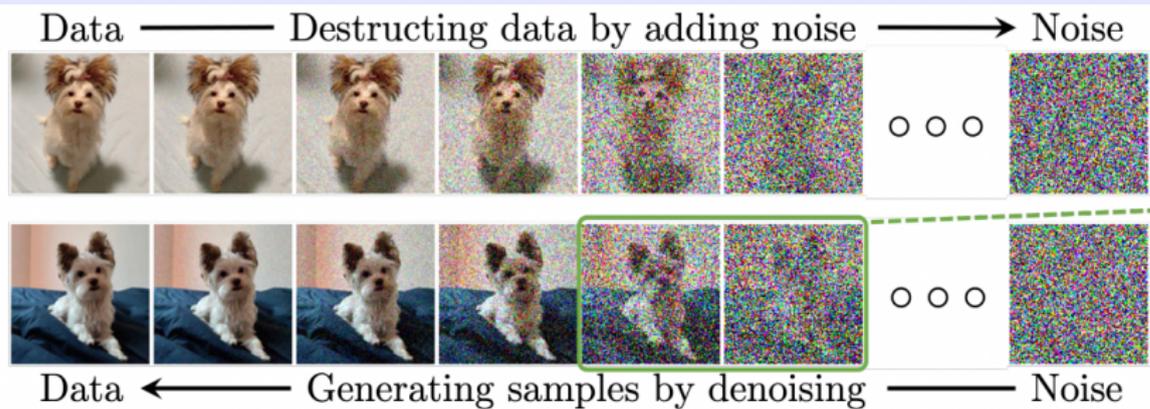
$$Z_t = \sqrt{\alpha_t} X + \sqrt{1 - \alpha_t} \epsilon_t \quad (14)$$



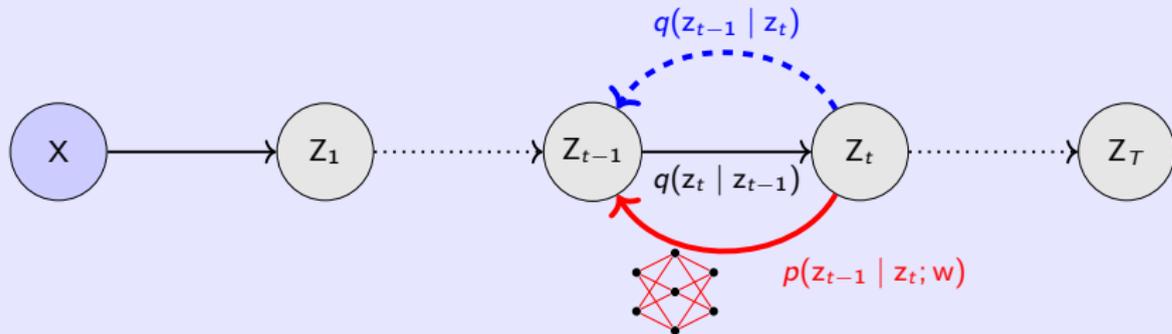
Forward/Reverse Process



Forward/Reverse Process

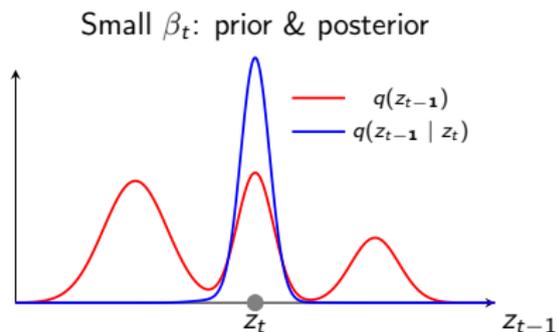
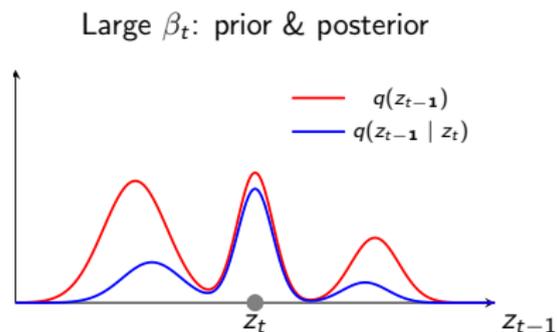
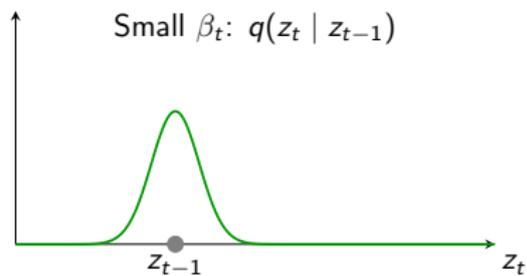
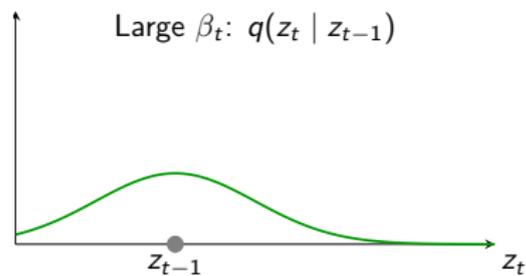
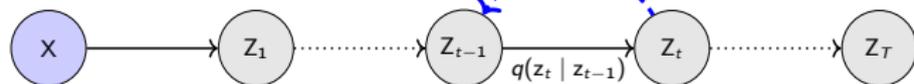


Probabilistic Graphical Model

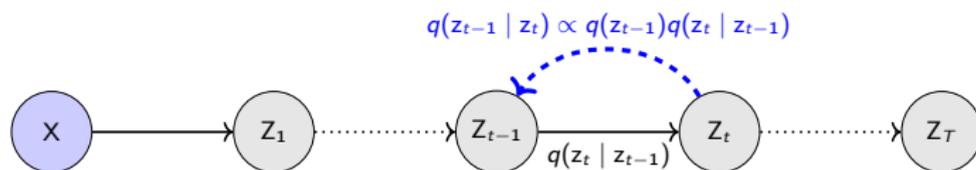


Reverse conditionals

$$q(z_{t-1} | z_t) \propto q(z_{t-1})q(z_t | z_{t-1})$$



Computing $\mathbb{E}[Z_{t-1}|z_t]$



1. Compute $\mathbb{E}[Z_{t-1}|z_t, x] = \frac{\sqrt{\alpha_{t-1}}\beta_t x + (1-\alpha_{t-1})\sqrt{1-\beta_t} z_t}{1-\alpha_t}$

2. Use iterated expectation:

$$\mathbb{E}[Z_{t-1}|z_t] = \mathbb{E}_{X|Z_t} \mathbb{E}_{Z_{t-1}|X, Z_t} [Z_{t-1}|Z_t, X] \quad (15)$$

$$= \frac{\sqrt{\alpha_{t-1}}\beta_t \mathbb{E}[X | z_t] + (1 - \alpha_{t-1})\sqrt{1 - \beta_t} z_t}{1 - \alpha_t} \quad (16)$$

Tweedie's Formula and the Score Function

$$Z_t = \sqrt{\alpha_t} X + \sqrt{1 - \alpha_t} \epsilon \quad (17)$$

$$\mathbb{E}[X|z_t] = \frac{1}{\sqrt{\alpha_t}} [z_t + (1 - \alpha_t) \nabla_{z_t} \log f(z_t)] \quad (18)$$

$$= \frac{z_t}{\sqrt{\alpha_t}} - \frac{\sqrt{(1 - \alpha_t)}}{\sqrt{\alpha_t}} \mathbb{E}[\epsilon|z_t] \quad (19)$$

► $\nabla_{z_t} \log f(z_t)$ is the (Stein) score function

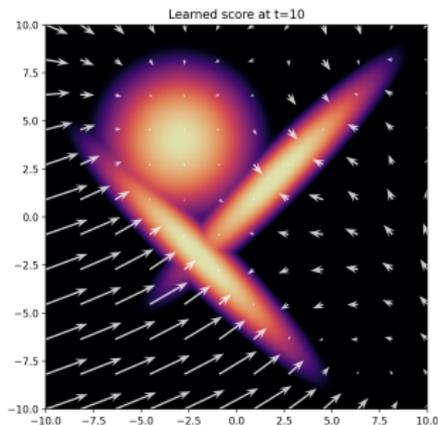
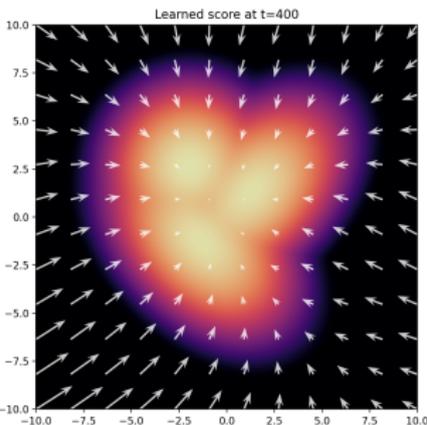
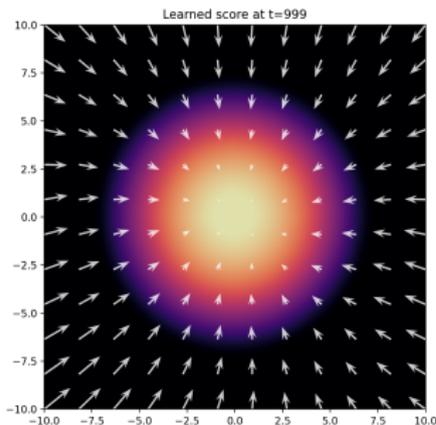
Tweedie's Formula and the Score Function

$$Z_t = \sqrt{\alpha_t} X + \sqrt{1 - \alpha_t} \epsilon \quad (17)$$

$$\mathbb{E}[X|z_t] = \frac{1}{\sqrt{\alpha_t}} [z_t + (1 - \alpha_t) \nabla_{z_t} \log f(z_t)] \quad (18)$$

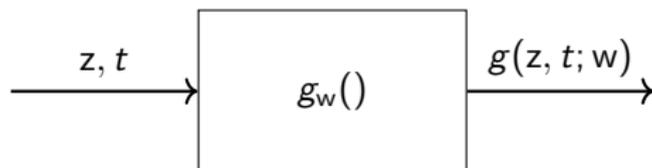
$$= \frac{z_t}{\sqrt{\alpha_t}} - \frac{\sqrt{(1 - \alpha_t)}}{\sqrt{\alpha_t}} \mathbb{E}[\epsilon|z_t] \quad (19)$$

► $\nabla_{z_t} \log f(z_t)$ is the (Stein) score function



Data Driven method to estimate $g(z)$ using a NN

1. Draw a batch of samples $(x^1, \epsilon^1), (x^2, \epsilon^2), \dots, (x^B, \epsilon^B)$, $X^i \sim q_X(x)$, $\epsilon_t^i \sim q_\epsilon()$
2. Draw $t \sim U(0, T]$
3. Form $z^i = \sqrt{\alpha_t} x^i + \sqrt{1 - \alpha_t} \epsilon^i$
4. Train $g(z, t; w)$ to minimize $\mathcal{L}(w) := \frac{1}{B} \sum_i \|\epsilon^i - g(z^i, t; w)\|^2$



Key points:

- ▶ Produces an approximation to the conditional expectation $\mathbb{E}[\epsilon|z]$
- ▶ It only requires a dataset of (x, t, ϵ, z) quadruplets
- ▶ Minimize quadratic loss if the goal is to estimate a conditional mean

Computing $\mathbb{E}[Z_{t-1}|z_t]$

$$\mathbb{E}[Z_{t-1}|z_t] = \frac{z_t}{\sqrt{1-\beta_t}} - \frac{\beta_t}{\sqrt{1-\beta_t}} \frac{1}{\sqrt{1-\alpha_t}} \mathbb{E}[\epsilon_t|z_t] \quad (20)$$

Approximate the function $\mathbb{E}[\epsilon_t|z_t]$ for every t by using a neural network $g(t, z_t, w)$ and compute $\mathbb{E}[Z_{t-1}|z_t]$ as

$$\mathbb{E}[Z_{t-1}|z_t] = \frac{z_t}{\sqrt{1-\beta_t}} - \frac{\beta_t}{\sqrt{1-\beta_t}} \frac{1}{\sqrt{1-\alpha_t}} g(t, z_t, w) \quad (21)$$

$$= \mathcal{D}(t, z_t) \quad (22)$$

Training Denoising Diffusion Probabilistic Models (DDPM)

Algorithm 20.1 C. M. Bishop and H. Bishop 2023: Training a denoising diffusion probabilistic model

Require: Training data $\mathcal{D} = \{x_n\}$, Noise schedule $\{\alpha_1, \dots, \alpha_T\}$

Ensure: Network parameters w

1: **repeat**

2: $x^1, \dots, x^B \sim \mathcal{D}$ ▷ Sample a mini-batch of data

3: $t^1, \dots, t^B \sim \{1, \dots, T\}$ ▷ Sample points along the Markov chain

4: $\epsilon^1, \dots, \epsilon^B \sim \mathcal{N}(0, I)$ ▷ Sample noise vectors

5: $z^i \leftarrow \sqrt{\alpha_{t^i}} x^i + \sqrt{1 - \alpha_{t^i}} \epsilon^i$ ▷ Evaluate noisy latent variable

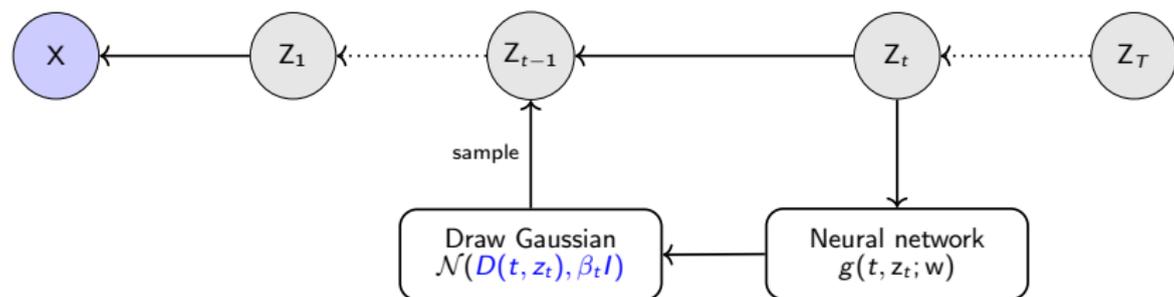
6: $\mathcal{L}(w) \leftarrow \frac{1}{B} \sum_i \|g(t^i, z^i, w) - \epsilon^i\|^2$ ▷ Compute loss term

7: Take optimizer step

8: **until** converged

9: **return** w

Stochastic Sampling in DDPM

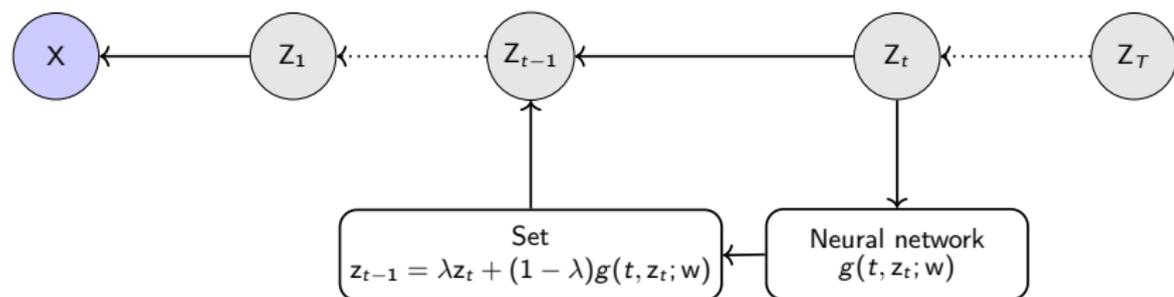


- ▶ When $\beta_t \rightarrow 0$, there is a theoretical guarantee of sampling from $q_X(x)$
- ▶ Information-theoretic proof using I-MMSE by Reeves and Pfister¹
- ▶ The structure of $g(t, z_t; w)$ matters (U-net for images)
- ▶ Working in the angular domain allows the use of a simple CNN²

¹G. Reeves and H. D. Pfister (2025). "Information-theoretic proofs for diffusion sampling". In: *2025 IEEE International Symposium on Information Theory (ISIT)*. IEEE, pp. 1–6.

²B. Fesl et al. (2024). "Diffusion-Based Generative Prior for Low-Complexity MIMO Channel Estimation". In: *IEEE Wireless Communications Letters* 13.12, pp. 3493–3497. DOI: 10.1109/LWC.2024.3474570.

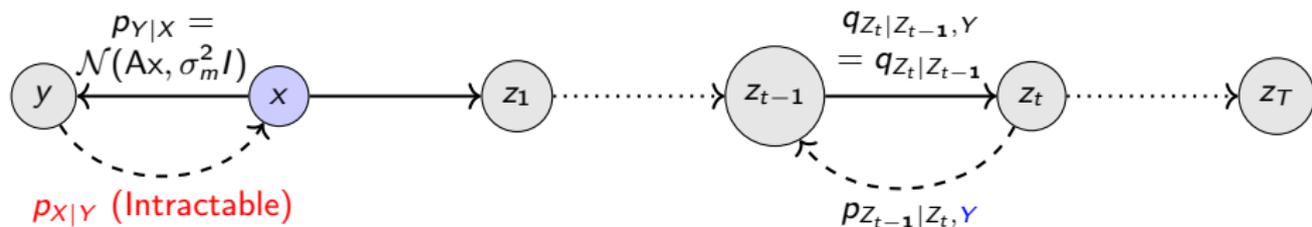
Deterministic Sampling in DDIM



- ▶ z_{t-1} is a deterministic function of z_t

Solving Inverse problems using Diffusion Models³

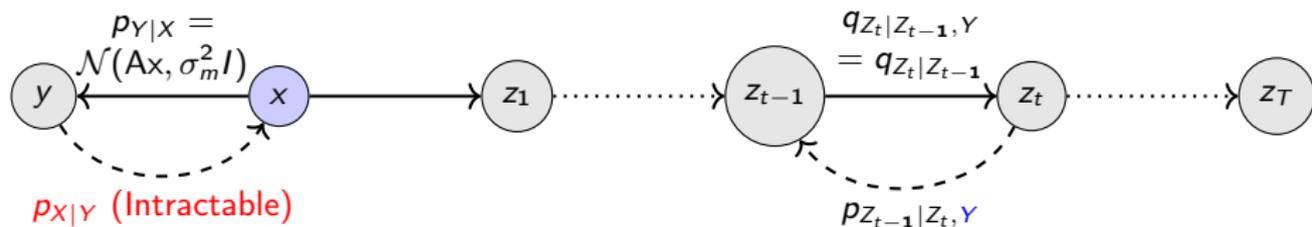
- Inverse problem: Estimate x given $y = Ax + n$, n is Gaussian noise.



³G. Daras et al. (2024). "A survey on diffusion models for inverse problems". In: *arXiv preprint arXiv:2410.00083*.

Solving Inverse problems using Diffusion Models³

- ▶ Inverse problem: Estimate x given $y = Ax + n$, n is Gaussian noise.

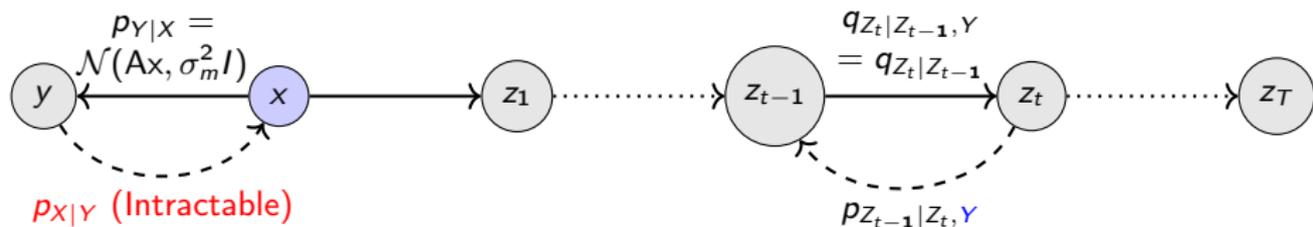


- ▶ Conditional score: $\nabla_{z_t} \log p(z_t|y) = \underbrace{\nabla_{z_t} \log p(z_t)}_{\text{pre-training}} + \underbrace{\nabla_{z_t} \log p(y|z_t)}_{\text{guidance}}$

³G. Daras et al. (2024). "A survey on diffusion models for inverse problems". In: *arXiv preprint arXiv:2410.00083*.

Solving Inverse problems using Diffusion Models³

- ▶ Inverse problem: Estimate x given $y = Ax + n$, n is Gaussian noise.

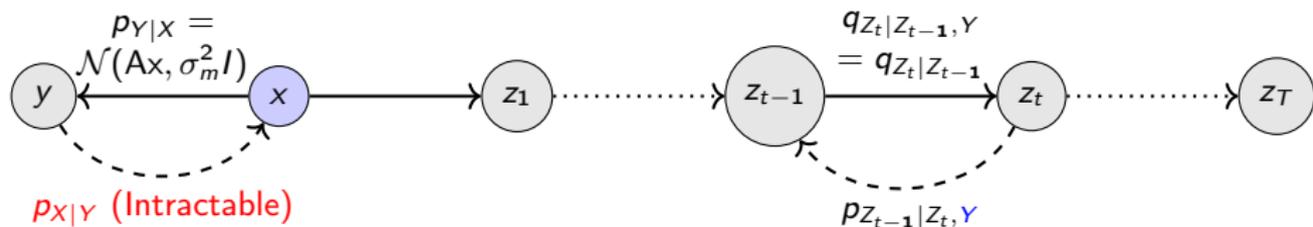


- ▶ Conditional score: $\nabla_{z_t} \log p(z_t|y) = \underbrace{\nabla_{z_t} \log p(z_t)}_{\text{pre-training}} + \underbrace{\nabla_{z_t} \log p(y|z_t)}_{\text{guidance}}$
- ▶ Approximation: $p_{Y|z_t}(y | z_t) \approx p_{Y|X}(y | \mathbb{E}[X | z_t])$
- ▶ Tweedie's formula: $\sqrt{\alpha_t} \mathbb{E}[X | z_t] = z_t + (1 - \alpha_t) \nabla_{z_t} \log p_{z_t}(z_t)$.

³G. Daras et al. (2024). "A survey on diffusion models for inverse problems". In: *arXiv preprint arXiv:2410.00083*.

Solving Inverse problems using Diffusion Models³

- ▶ Inverse problem: Estimate x given $y = Ax + n$, n is Gaussian noise.

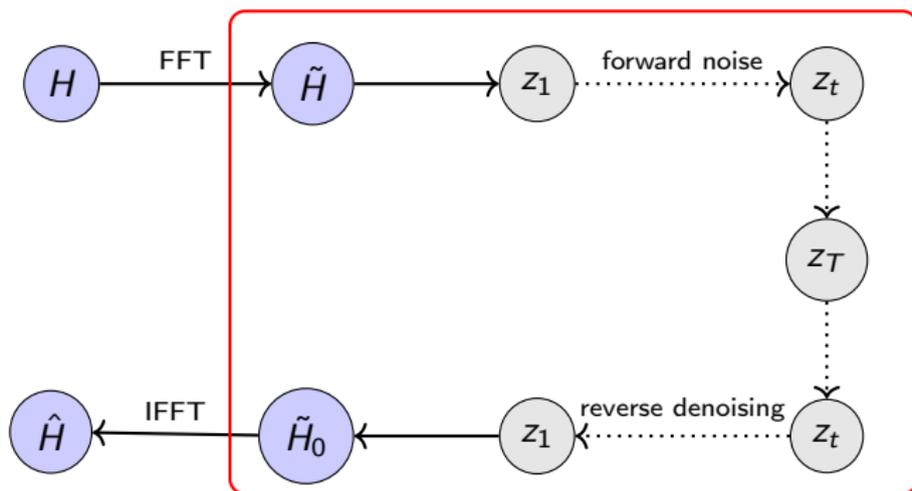


- ▶ Conditional score: $\nabla_{z_t} \log p(z_t|y) = \underbrace{\nabla_{z_t} \log p(z_t)}_{\text{pre-training}} + \underbrace{\nabla_{z_t} \log p(y|z_t)}_{\text{guidance}}$
- ▶ Approximation: $p_{Y|z_t}(y | z_t) \approx p_{Y|X}(y | \mathbb{E}[X | z_t])$
- ▶ Tweedie's formula: $\sqrt{\alpha_t} \mathbb{E}[X | z_t] = z_t + (1 - \alpha_t) \nabla_{z_t} \log p_{z_t}(z_t)$.
- ▶ *Denoising is a special case, Blind estimation is a generalization*

³G. Daras et al. (2024). "A survey on diffusion models for inverse problems". In: *arXiv preprint arXiv:2410.00083*.

Angular-Domain Representation for Efficient Diffusion Modeling

Diffusion Model operates in angular domain



SNR-Matched Initialization and Pilot Preprocessing

1) Pilot Decorrelation and Angular Transformation

$$Y_p = HS_p + W_p$$

$$Y_p S_p^H = H + W_p S_p^H$$

$$\tilde{Y}_p = \tilde{H} + \tilde{W}_p$$

- ▶ Orthogonal pilots remove pilot dependence
- ▶ Channel estimation becomes a **denoising problem** in angular domain

SNR-Matched Initialization and Pilot Preprocessing

1) Pilot Decorrelation and Angular Transformation

$$\begin{aligned}Y_p &= HS_p + W_p \\ Y_p S_p^H &= H + W_p S_p^H \\ \tilde{Y}_p &= \tilde{H} + \tilde{W}_p\end{aligned}$$

- ▶ Orthogonal pilots remove pilot dependence
- ▶ Channel estimation becomes a **denoising problem** in angular domain

2) Variance Normalization

$$\frac{1}{\sqrt{1+\sigma^2}} \tilde{Y}_p = \frac{1}{\sqrt{1+\sigma^2}} \tilde{H} + \frac{\sigma}{\sqrt{1+\sigma^2}} W$$

- ▶ Matches variance-preserving diffusion forward process

SNR-Matched Initialization and Pilot Preprocessing

1) Pilot Decorrelation and Angular Transformation

$$\begin{aligned} Y_p &= HS_p + W_p \\ Y_p S_p^H &= H + W_p S_p^H \\ \tilde{Y}_p &= \tilde{H} + \tilde{W}_p \end{aligned}$$

- ▶ Orthogonal pilots remove pilot dependence
- ▶ Channel estimation becomes a **denoising problem** in angular domain

2) Variance Normalization

$$\frac{1}{\sqrt{1+\sigma^2}} \tilde{Y}_p = \frac{1}{\sqrt{1+\sigma^2}} \tilde{H} + \frac{\sigma}{\sqrt{1+\sigma^2}} W$$

- ▶ Matches variance-preserving diffusion forward process

3) SNR-Matched Reverse Initialization

$$\text{SNR}_{\text{DM}}(t) = \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}$$

$$t^* = \arg \min_t \left| \text{SNR}(\tilde{Y}) - \text{SNR}_{\text{DM}}(t) \right|$$

Gram Matrix Estimation from Data Symbols

Data observation:

$$Y_d = HS_d + W_d.$$

Sample Gram matrix:

$$\frac{1}{N_d} Y_d Y_d^H = \frac{1}{N_d} (HS_d + W_d)(HS_d + W_d)^H$$

▶ With sufficiently long data block and approximately uncorrelated symbols:

$$\frac{1}{N_d} S_d S_d^H \approx I, \quad \text{cross terms vanish.}$$

▶ Hence

$$\frac{1}{N_d} Y_d Y_d^H \approx HH^H + \sigma_d^2 I.$$

Gram estimate:

$$\hat{R} = \frac{1}{N_d} Y_d Y_d^H - \sigma_d^2 I.$$

Angular-domain Gram:

$$\hat{\tilde{R}} = \Phi_{N_R} \hat{R} \Phi_{N_R}^T.$$

Likelihood & Gram Guidance: Joint Reverse Update

Gram guidance.

$$f(\tilde{H}_t) = - \left\| \tilde{H}_t \tilde{H}_t^H - \hat{R} \right\|_F^2, \quad \mathbf{g}_{\text{Gram}}(\tilde{H}_t) = 4(\hat{R} - \tilde{H}_t \tilde{H}_t^H) \tilde{H}_t.$$

Likelihood guidance. Use the denoised estimate $\mathcal{T}(\tilde{H}_t) = \mathbb{E}[\tilde{H}_0 \mid \tilde{H}_t]$:

$$\mathbf{g}_{\text{like}}(\tilde{H}_t) \triangleq \nabla_{\tilde{H}_t} \log p(\tilde{Y} \mid \tilde{H}_t) \approx \frac{1}{\sigma^2} (\tilde{Y} - \mathcal{T}(\tilde{H}_t)).$$

Joint guided reverse step:

$$\tilde{H}_{t-1} = \mathcal{D}_t(\tilde{H}_t) + \lambda_{\text{like},t} \mathbf{g}_{\text{like}}(\tilde{H}_t) + \lambda_{\text{Gram},t} \mathbf{g}_{\text{Gram}}(\tilde{H}_t).$$

Algorithm: Guided Diffusion Channel Estimation

Inputs: $Y_p, Y_d, S_p, \{\mathcal{D}_t\}_{t=1}^T$

Step 1: Pilot Preprocessing (Angular Domain)

$$\tilde{Y} = \Phi_{N_R} Y_p S_p^H (S_p S_p^H)^{-1} \Phi_{N_T}^T$$

Step 2: SNR Matching Initialization

$$t^* = \arg \min_t |\text{SNR}(\tilde{Y}) - \text{SNR}_{DM}(t)|$$

$$\tilde{H}_{t^*} = (1 + \sigma^2)^{-1/2} \tilde{Y}$$

Step 3: Gram Estimation from Data

$$\hat{R} = \frac{1}{N_d} Y_d Y_d^H - \sigma_d^2 I$$

$$\hat{\hat{R}} = \Phi_{N_R} \hat{R} \Phi_{N_R}^T$$

Algorithm: Guided Diffusion Channel Estimation

Step 4: Guided Reverse Diffusion

For $t = t^*, \dots, 1$:

$$\mathbf{g}_{\text{Gram}} = 4(\hat{\mathbf{R}} - \tilde{\mathbf{H}}_t \tilde{\mathbf{H}}_t^H) \tilde{\mathbf{H}}_t$$

$$\mathbf{g}_{\text{like}} = \frac{1}{\sigma^2} (\tilde{\mathbf{Y}} - \mathcal{T}(\tilde{\mathbf{H}}_t))$$

$$\tilde{\mathbf{H}}_{t-1} = \mathcal{D}_t(\tilde{\mathbf{H}}_t) + \lambda_{\text{Gram},t} \mathbf{g}_{\text{Gram}} + \lambda_{\text{like},t} \mathbf{g}_{\text{like}}$$

Step 5: Transform Back to Spatial Domain

$$\hat{\mathbf{H}} = \Phi_{N_R}^H \tilde{\mathbf{H}}_0 \Phi_{N_T}^*$$

Channel Estimation Performance⁴

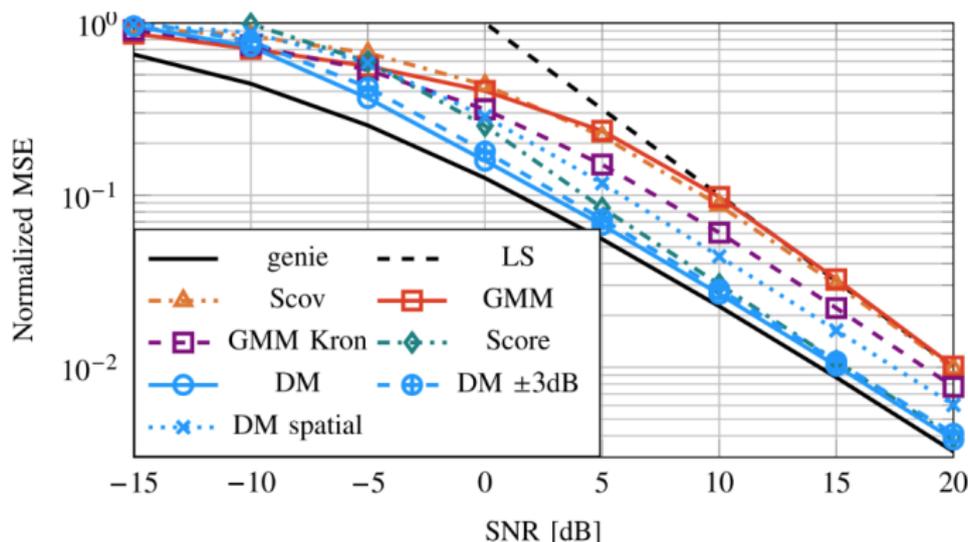
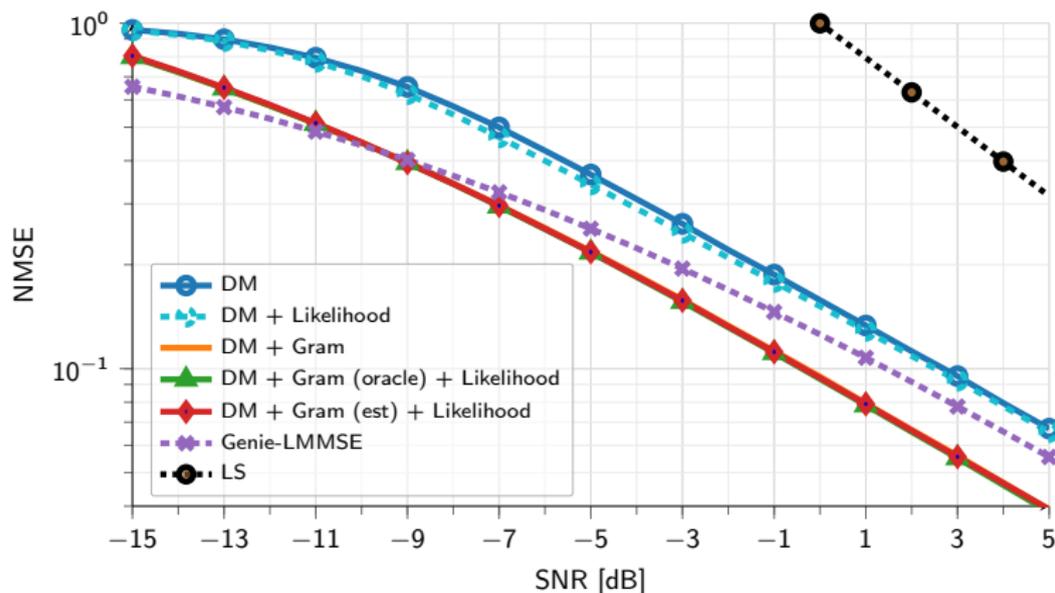


Fig. 2. MSE performance for the 3GPP channel model with three propagation clusters and $L = 100$ DM steps.

► Diffusion prior based estimation outperforms baselines

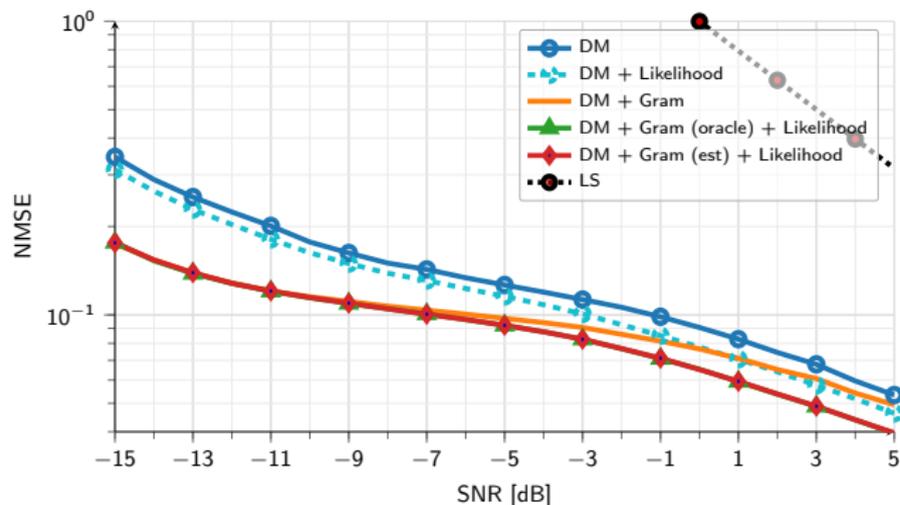
⁴B. Fesl et al. (2024). “Diffusion-Based Generative Prior for Low-Complexity MIMO Channel Estimation”. In: *IEEE Wireless Communications Letters* 13.12, pp. 3493–3497. DOI: 10.1109/LWC.2024.3474570.

3GPP Results: Normalized MSE vs. SNR



- ▶ Significant (3-4dB) improvement over Diffusion even at high SNRs

QuaDRiGa Results: NMSE vs. SNR



- ▶ **Gram guidance** again yields a large gain over DM.
- ▶ **Likelihood contribution** is more pronounced than in 3GPP.

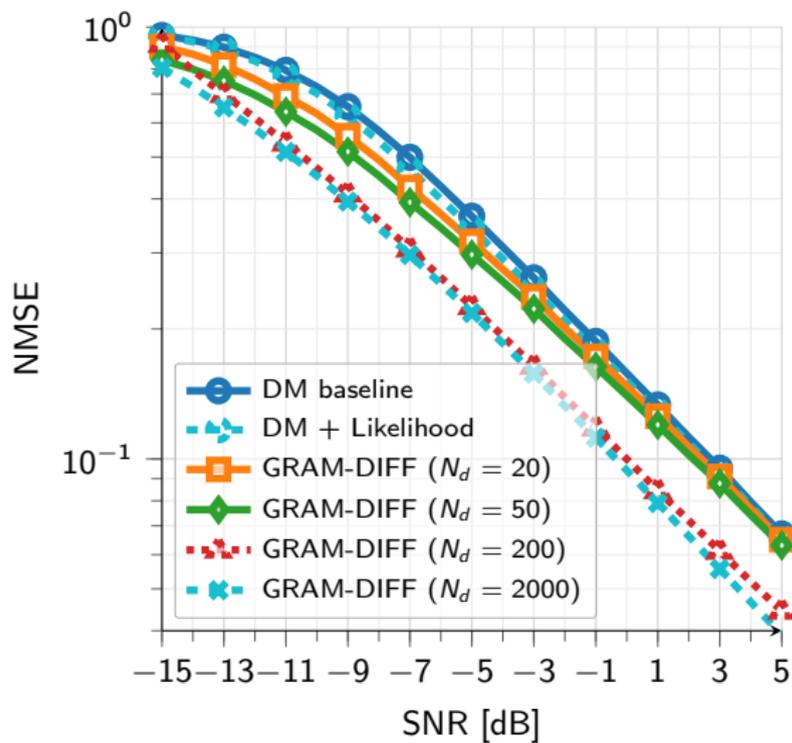
Complexity comparison⁵ for $N_R = 64, N_T = 16$

Method	Parameters	Online Complexity
Scov	1.05×10^6	$\mathcal{O}(N_R N_T^2)$
GMM	1.35×10^8	$\mathcal{O}(K N_R N_T^2)$
GMM Kron	7.22×10^4	$\mathcal{O}(K N_R N_T^2)$
Score	5.89×10^6	$\mathcal{O}(L_{Sc} k^2 D J C_{\max}^2 N_R N_T)$
Diffusion	5.50×10^4	$\mathcal{O}(N_R N_T (\hat{\ell} k^2 C_{\max}^2 + \log N_R N_T))$
GRAM-Diff		$\mathcal{O}(N_R N_T (\hat{\ell} (k^2 C_{\max}^2 + N_R) + \log N_R N_T))$

- ▶ Diffusion in the angular domain allows the use of a simple CNN (Fesl et al)
- ▶ Extra complexity due to Gram guidance is insignificant

⁵B. Fesl et al. (2024). “Diffusion-Based Generative Prior for Low-Complexity MIMO Channel Estimation”. In: *IEEE Wireless Communications Letters* 13.12, pp. 3493–3497. DOI: 10.1109/LWC.2024.3474570.

Coherence-Time Robustness: Graceful Degradation



Key observations.

- ▶ Smaller $N_d \Rightarrow$ noisier Gram estimate.
- ▶ Reduced guidance strength leads to gradual performance degradation.
- ▶ Performance smoothly interpolates toward **DM+Likelihood**.
- ▶ No abrupt failure under severe sample limitation.
- ▶ GRAM-DIFF remains effective even for very small N_d .

Alternative Low-Complexity View and Future Work

$$\tilde{\mathbf{H}} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^H$$

- ▶ Since $\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H = \tilde{\mathbf{U}}\tilde{\Sigma}^2\tilde{\mathbf{U}}^H$, we can get $\tilde{\mathbf{U}}, \tilde{\Sigma}$ when $\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H$ is estimated
- ▶ The remaining uncertainty is only in $\tilde{\mathbf{V}}$

⁶A. K. Jagannatham and B. D. Rao (2006). “Whitening-rotation-based semi-blind MIMO channel estimation”. In: *IEEE Transactions on Signal Processing* 54.3, pp. 861–869.

Alternative Low-Complexity View and Future Work

$$\tilde{\mathbf{H}} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^H$$

- ▶ Since $\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H = \tilde{\mathbf{U}}\tilde{\Sigma}^2\tilde{\mathbf{U}}^H$, we can get $\tilde{\mathbf{U}}, \tilde{\Sigma}$ when $\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H$ is estimated
- ▶ The remaining uncertainty is only in $\tilde{\mathbf{V}}$

Current pipeline

- ▶ Generate full channel $\tilde{\mathbf{H}}$
- ▶ Gram-guided diffusion
- ▶ Per-step complexity: $\mathcal{O}(N_R^2 N_T)$

Eigen-domain pipeline

- ▶ Transform to eigen-domain
- ▶ Learn only $\tilde{\mathbf{V}}$
- ▶ Reduced dimension when $N_T < N_R$

Future direction: low-complexity diffusion via eigen-domain generation

- ▶ This was noted by Jagannatham and Rao⁶ (Not for Diffusion models)

⁶A. K. Jagannatham and B. D. Rao (2006). “Whitening-rotation-based semi-blind MIMO channel estimation”. In: *IEEE Transactions on Signal Processing* 54.3, pp. 861–869.

Conclusion

- ▶ Reviewed training and generation using Diffusion models
- ▶ Extended Fest et al's algorithm with Gram and Likelihood guidance
- ▶ Diffusion models provide significant improvement over other baselines
- ▶ Gram guidance provides addition 3-4 dB improvement for 64x12 MIMO
- ▶ More details can be found in⁷

⁷X. Wang and K. Narayanan (2025). "GRAM-DIFF: Gram Matrix Guided Diffusion for MIMO Channel Estimation". In: *arXiv preprint arXiv:2602.15187*.