# Computational Mathematics and AI

## Lecture 9: Machine Learning for Inverse Problems

Lars Ruthotto
Departments of Mathematics and Computer Science
lruthotto@emory.edu
larsruthotto

slido.com #CBMS25

# Reading List

**Historical Context:** Inverse problems require regularization due to ill-posedness. Recent advances use diffusion priors and simulation-based inference.

**Key Readings:**

1. Antun et al. (2020) – On Instabilities of Deep Learning in Image Reconstruction. *PNAS*

   Critical analysis of neural network stability in inverse problems.

2. Chung et al. (2023) – Diffusion Posterior Sampling for Inverse Problems. *ICLR*

   Current state-of-the-art using diffusion priors.

3. Cranmer et al. (2020) – The Frontier of Simulation-Based Inference. *PNAS*

   Likelihood-free Bayesian inference methods.

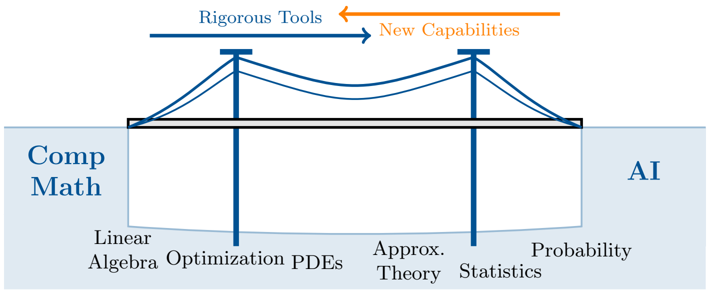
4. Wang et al. (2023) – Efficient Neural Approaches for Conditional OT. arXiv:2310.16975

   COT-Flow for fast posterior estimation.

5. Kawar et al. (2022) – Denoising Diffusion Restoration Models. *NeurIPS*
   DDRM for linear inverse problems.

**Outline:** Ill-posedness $\rightarrow$ Bayesian $\rightarrow$ Simulation-Based Inference $\rightarrow$ Diffusion

# Roadmap & Learning Objectives




## Learning Objectives

1. Why direct neural networks fail catastrophically
2. Simulation-based inference for expensive simulators
3. Diffusion models as priors

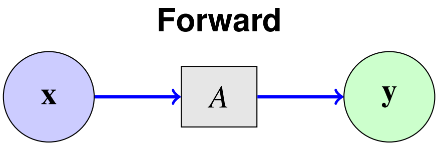
## Roadmap

1. Inverse problems background
2. NN failure modes
3. Simulation-based inference (SBI)
4. Diffusion posterior sampling

**Today: AI provides new opportunities for inverse problems**

# Ill-posed Inverse Problems

# Forward vs. Inverse Problems




**Forward Problem**:

- Given parameters $\mathbf{x} \in \mathbb{R}^n$
- Predict observations $\mathbf{y} = A(\mathbf{x}) + \epsilon$
- Well-posed, tractable

**Inverse Problem**:

- Given observations $\mathbf{y} \in \mathbb{R}^m$
- Infer parameters $\mathbf{x}$
- Mathematically treacherous

## Hadamard Well-Posedness Conditions

✓ **Existence**: Solution exists for all data

✓ **Uniqueness**: Solution is unique

X **Stability**: Small data changes $\rightarrow$ small solution changes

**ill-posed inverse problems are ubiquitous and known to be challenging**

# Motivating Example: Image Deblurring

Noisy Data

## Key Observation

- Least squares solution amplifies noise (ill-posedness!)
- Early stopping provides implicit regularization
- Continuing optimization makes things **worse**

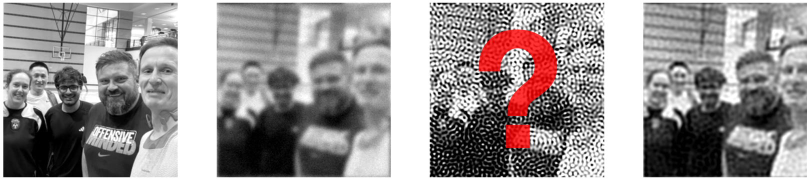
## This Lecture

- Why direct neural networks fail catastrophically
- Generative AI for inverse problems

**early stopping is implicit regularization for ill-posed problems**

# The Naive Approach: Directly Learning $A^{-1}$

Ground Truth | Noisy Data | NN Recon | Early Stopped



**Training Setup**:

$(\mathbf{x}_i, \mathbf{y}_i)$

↓

$G_\theta(\mathbf{y})$

↓

$L = \mathbb{E}[\|\mathbf{x} - G_\theta(\mathbf{y})\|^2]$

Network learns: $G_\theta \approx A^{-1}$

**Why This Fails**:

- The inverse $A^{-1}$ is **unbounded** (ill-posed!)
- $\Rightarrow$ Network inherits unbounded Lipschitz constant
- **Not a deep learning artifact**—this is *linear algebra*
- Even implicit regularization cannot overcome $\|A^{-1}\| = \infty$
- More data/better net $\rightarrow$ better fit to **unstable map**

**Key Insight** (Hansen et al., 2021): Deep learning becomes unstable. Training on clean pairs teaches the network an unstable inverse map.

# Takeaway from Direct Learning Failure

## Three Critical Lessons

X **Direct inverse learning does not overcome ill-posedness**

- Network amplifies noise
- Training on clean data $\neq$ robustness to noise

X **No amount of data or architecture fixes instability**

- More capacity $\rightarrow$ worse instability (closer to $A^{-1}$)
- Better data $\rightarrow$ better fit to unstable map
- This is not a typical ML problem

✓ **We need methods that acknowledge uncertainty**

- Single point estimate is insufficient
- Multiple solutions may fit data equally well
- Uncertainty quantification is essential

# Bayesian Framework

# The Bayesian Formulation

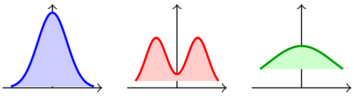
## Posterior Distribution

$$\pi(\mathbf{x}|\mathbf{y}) \propto \underbrace{p(\mathbf{y}|\mathbf{x})}_{\text{likelihood}} \cdot \underbrace{\pi(\mathbf{x})}_{\text{prior}}$$

**Components**:

- **Likelihood** $p(\mathbf{y}|\mathbf{x})$: Forward model + noise
  - Gaussian: $p(\mathbf{y}|\mathbf{x}) \propto \exp(-\frac{1}{2\sigma^2}\|A(\mathbf{x}) - \mathbf{y}\|^2)$
- **Prior** $\pi(\mathbf{x})$: What we (think we) know before seeing data
- **Posterior** $\pi(\mathbf{x}|\mathbf{y})$: Updated belief given data

## Posteriors Reveal Structure




- **Multi-modality**: Multiple valid solutions
- **Width**: Uncertainty in inference
- **Shape**: Guides decision-making

## Advantages of Full Posteriors Over Just Point Estimates

- Ill-posedness $\rightarrow$ non-uniqueness: multiple solutions fit data equally well
- Posterior width quantifies reliability; shapes guide different decisions

**uncertainty is information, not failure—the full posterior matters**

# Classical Posterior Sampling Methods

**MCMC (Markov Chain Monte Carlo)**:

- **Algorithm**: Construct Markov chain with stationary distribution $\pi(\mathbf{x}|\mathbf{y})$
- **Variants**: Metropolis-Hastings, Gibbs, HMC (Hamiltonian)
- **Cost**: $10^4$–$10^6$ likelihood evaluations
- **Assumes**: Can evaluate $p(\mathbf{y}|\mathbf{x})$ pointwise

✓ Asymptotically exact

– Slow mixing in high dimensions

– Each step needs likelihood

**Variational Inference (VI)**:

- **Algorithm**: Optimize $q_\varphi(\mathbf{x}) \approx \pi(\mathbf{x}|\mathbf{y})$ in parametric family
- **Objective**: Minimize reverse KL $\mathrm{KL}(q\|\pi)$ via ELBO
- **Cost**: $10^3$–$10^5$ gradient steps
- **Assumes**: Tractable family $q_\varphi$

– Optimization

– Approximation bias (may miss modes)

– Still needs likelihood access

**What if we can't evaluate the likelihood?**

Forward model $A$ is a black-box simulator (climate, turbulence, PDEs)
Each simulation: minutes to hours $\rightarrow$ MCMC/VI infeasible

# Connection to Lecture 6: Generative Modeling

## Recall Lecture 6

- Learned distributions of natural signals using diffusion models
- Score-based methods via Fokker-Planck PDE
- Generated samples from $p(\mathbf{x})$ (unconditional)

## Lecture 9: Conditional Generation

- Use learned priors to solve inverse problems
- Sample from posterior $\pi(\mathbf{x}|\mathbf{y})$ (conditional on observations)
- Quantify uncertainty in reconstructions

## Key Equation: Posterior Score Decomposition

$$\underbrace{\nabla_{\mathbf{x}} \log \pi(\mathbf{x}|\mathbf{y})}_{\text{posterior score}} = \underbrace{\nabla_{\mathbf{x}} \log \pi(\mathbf{x})}_{\text{prior (from L6)}} + \underbrace{\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})}_{\text{likelihood (from } A)}$$

## Modular Composition

- Pre-train prior $\pi(\mathbf{x})$ once on clean data
- Apply to many inverse problems with different forward models $A$
- No retraining needed for each new problem!

# Simulation-Based Inference

# Neural Posterior Estimation (NPE)

## Core Idea: Learn $\pi(\mathbf{x}|\mathbf{y})$ Directly

| 1. Simulate | | 2. Train | | 3. Infer |
|---|---|---|---|---|
| Sample $\mathbf{x}_i \sim \pi(\mathbf{x})$<br>Run $\mathbf{y}_i = A(\mathbf{x}_i) + \epsilon$<br>Collect pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ | $\longrightarrow$ | Neural density estimator<br>$q_\varphi(\mathbf{x}|\mathbf{y})$<br>(flow, score) | $\longrightarrow$ | Given new $\mathbf{y}_{\text{obs}}$<br>Sample from $q_\varphi(\mathbf{x}|\mathbf{y}_{\text{obs}})$<br>**Instantly!** |

## Amortized Inference

- **Upfront cost**: Run $N$ simulations, train neural network
- **Per-query cost**: Sample from $q_\varphi(\mathbf{x}|\mathbf{y})$ in milliseconds
- **Benefit**: Pay once, infer many times (different observations)

## Sequential NPE (SNPE) Papamakarios et al. 2019; Greenberg et al. 2019

- Iteratively refine: simulate from current posterior estimate, retrain
- Converges in 3–5 rounds; 100–1000$\times$ fewer simulations than MCMC
- Focuses computational budget on high-posterior regions

# COT-Flow: Optimal Transport for Posterior Estimation

**Goal:** Learn posterior $\pi(\mathbf{x}|\mathbf{y})$ from samples $(\mathbf{x}, \mathbf{y}) \sim \pi(\mathbf{x}, \mathbf{y})$

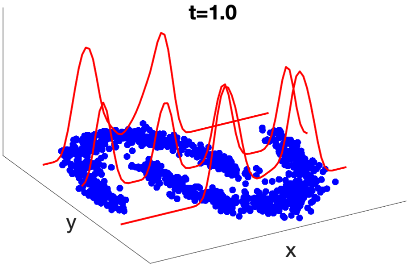
**Recall from Lecture 6**: OT-Flow learns generator $g_\theta : \mathcal{Z} \to \mathcal{X}$ via

- Neural ODE: $\frac{d\mathbf{u}}{dt} = v_\theta(t, \mathbf{u})$
- OT regularization: minimize kinetic energy
- Conservative velocity: $v_\theta = -\nabla \Phi_\theta$

**Extension to Conditional Distributions:**
Train **conditional** generator $g_\theta(\mathbf{z}, \mathbf{y})$ such that

$$\pi(\mathbf{x}|\mathbf{y}) \approx \rho_Z(g_\theta^{-1}(\mathbf{x}, \mathbf{y})) \cdot |\det \nabla_{\mathbf{x}} g_\theta^{-1}(\mathbf{x}, \mathbf{y})|$$

**Idea**: condition velocity field on observation $\mathbf{y}$



Transport from prior $\rho_Z$ to posterior $\pi(\mathbf{x}|\mathbf{y})$

**COT-Flow: likelihood-free VI via OT-regularized flows**

# COT-Flow: Mean Field Game Formulation

**Training Objective** (OT-penalized maximum likelihood) Wang et al. 2023

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\pi} \left[ \underbrace{-\log q_\theta(\mathbf{x}|\mathbf{y})}_{\text{posterior fit}} + \alpha \underbrace{\int_0^1 \frac{1}{2}\|v_\theta(t,\mathbf{p},\mathbf{y})\|^2 dt}_{\text{kinetic energy}} + \beta \underbrace{\left\| \partial_t \Phi + \frac{1}{2\alpha}\|\nabla_x \Phi\|^2 \right\|^2}_{\text{HJB penalty}} \right]$$

**Hamilton-Jacobi-Bellman Equations** (from optimal control)

$$\partial_t \Phi(t,\mathbf{x},\mathbf{y}) - \frac{1}{2\alpha}\|\nabla_x \Phi(t,\mathbf{x},\mathbf{y})\|^2 = 0$$

**Implementation Strategy** (extends OT-Flow from Lecture 6)

1. Learn scalar potential $\Phi_\theta(t,\mathbf{x},\mathbf{y})$ with neural network
2. Use feedback form: $v_\theta(t,\mathbf{u},\mathbf{y}) = -\frac{1}{\alpha}\nabla_x \Phi_\theta(t,\mathbf{u},\mathbf{y})$
3. Explicit Laplacian: $\nabla \cdot v_\theta = -\frac{1}{\alpha}\Delta\Phi_\theta$ (efficient likelihood)
4. HJB penalty ($\beta > 0$) enforces optimality of transport paths

# COT-Flow: Stochastic Lotka-Volterra Example

## Problem Setup:

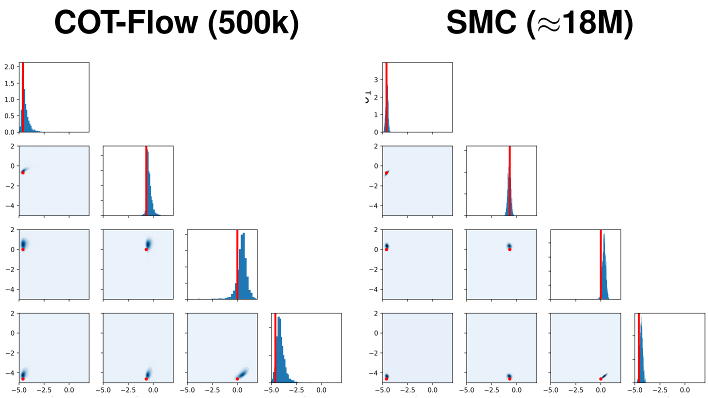
- Predator-prey dynamics
- $\mathbf{x} \in \mathbb{R}^4$: rate parameters
- $\mathbf{y} \in \mathbb{R}^9$: summary statistics

## Sample Efficiency:

- SMC: ≈6–18M simulations
- COT-Flow: 50k–500k sims
- **30–100× speedup**

## When to Use:

- Expensive PDE simulators
- Moderate dimensions ($d \leq 100$)
- Multi-modal posteriors




Posterior corner plots: diagonal = marginals, off-diagonal = 2D projections

**COT-Flow matches SMC quality with 30–100× fewer simulations**

# Summary: Bayesian Methods for Inverse Problems

## Three Approaches to Posterior Sampling

1. **Classical MCMC/VI**
   - Requires likelihood access: can evaluate $p(\mathbf{y}|\mathbf{x})$
   - Exact (asymptotically) but expensive: $10^5$–$10^6$ evaluations
   - Use when: likelihood cheap, moderate dimensions
2. **Simulation-Based Inference (SBI/NPE)**
   - Likelihood-free: only need simulator $\mathbf{y} = A(\mathbf{x}) + \epsilon$
   - Amortized: train once, infer many times instantly
   - Use when: expensive black-box simulators
3. **COT-Flow** (forward-KL VI, likelihood-free)
   - Best sample efficiency: 30–100$\times$ fewer simulations
   - Better mode coverage via optimal transport
   - Use when: very expensive PDEs, multi-modal posteriors

**Key Limitation**: All SBI methods learn a **specific** posterior $\pi(\mathbf{x}|\mathbf{y})$
If forward operator $A$ or prior $\pi(\mathbf{x})$ changes, **must retrain from scratch**

**next: diffusion models offer a modular alternative**

# Diffusion Models for Inverse Problems

# Score-Based Approach to Inverse Problems

## Bayes' Theorem Becomes Additive for Scores

$$\underbrace{\nabla_{\mathbf{x}} \log \pi(\mathbf{x}|\mathbf{y})}_{\text{posterior score}} = \underbrace{\nabla_{\mathbf{x}} \log \pi(\mathbf{x})}_{\text{prior score}} + \underbrace{\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})}_{\text{likelihood score}}$$

**Prior Score** (from L6):

- Learned by diffusion model: $s_\theta(\mathbf{x}, t)$
- Pre-trained once, reusable
- Encodes complex priors (natural images)

**Likelihood Score** (needed):

- Derived from forward model $A$
- Problem-specific
- Tweedie's formula

**Key Advantage over SBI/COT-Flow:** No retraining when $A$ changes—modular composition at inference

# Linear Gaussian Case: Exact Likelihood Score

**The Challenge** We need $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$, not at $t = 0$

At time $t$: $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon$ is noisy, but $\mathbf{y} = A\mathbf{x}_0 + \eta$ depends on clean $\mathbf{x}_0$

**Marginalize Over $\mathbf{x}_0$**

$$p(\mathbf{y}|\mathbf{x}_t) = \int p(\mathbf{y}|\mathbf{x}_0)\, p(\mathbf{x}_0|\mathbf{x}_t)\, d\mathbf{x}_0$$

**Key insight**: $p(\mathbf{x}_0|\mathbf{x}_t)$ is from the **Gaussian diffusion kernel** (not the prior!):

$$p(\mathbf{x}_0|\mathbf{x}_t) = \mathcal{N}\left(\frac{\mathbf{x}_t}{\alpha_t}, \frac{\sigma_t^2}{\alpha_t^2} I\right) \quad \text{(closed-form!)}$$

**Exact Score** Kawar et al. 2022: Gaussian convolution $\Rightarrow p(\mathbf{y}|\mathbf{x}_t) = \mathcal{N}(\mathbf{y}; \bar{\boldsymbol{\mu}}_t, \bar{\Sigma}_t)$

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \frac{1}{\alpha_t} A^\top \bar{\Sigma}_t^{-1} (\mathbf{y} - \bar{\boldsymbol{\mu}}_t) \quad \text{where } \bar{\boldsymbol{\mu}}_t = \frac{A\mathbf{x}_t}{\alpha_t},\ \bar{\Sigma}_t = \frac{\sigma_t^2}{\alpha_t^2} AA^\top + \sigma_\eta^2 I$$

**linear + Gaussian: exact closed-form; general: requires approximation (DPS)**

# Diffusion Posterior Sampling (DPS) Chung et al. 2023

## DPS Algorithm

**Input**: Observation $\mathbf{y}$, pre-trained score network $s_\theta(\mathbf{x}, t)$, forward model $A$

**For** $t = T, T-1, \ldots, 1$:

1. **Predict clean image via conditional expectation (Tweedie's formula)**:

$$\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = (\mathbf{x}_t + \sigma_t^2 s_\theta(\mathbf{x}_t, t))/\alpha_t$$

2. **Compute likelihood gradient**:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \approx -\nabla_{\mathbf{x}_t} \|A(\hat{\mathbf{x}}_0) - \mathbf{y}\|^2$$

3. **Posterior sampling step**:

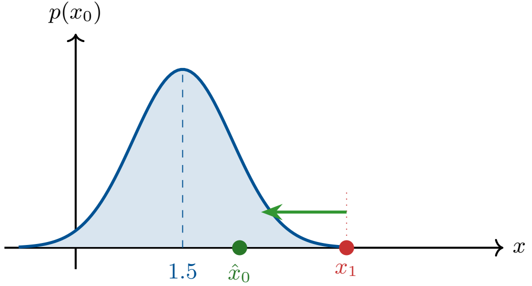
$$\mathbf{x}_{t-1} = \mu_t(\mathbf{x}_t) + \lambda \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) + g_t \epsilon$$

**Output**: Posterior sample $\mathbf{x}_0 \sim \pi(\mathbf{x}|\mathbf{y})$

# Tweedie's Formula: Score $\leftrightarrow$ Denoising

For noisy observation $\mathbf{x}_1 = \mathbf{x}_0 + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$ Efron 2011 :

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_1] = \mathbf{x}_1 + \sigma^2 \nabla_{\mathbf{x}_1} \log p(\mathbf{x}_1)$$




## Intuition: Denoising via the Score

- Score $\nabla \log p(\mathbf{x}_1)$ points toward high-density regions of the prior
- $\sigma^2$ scales the correction:
  - Large $\sigma$ (noisy) $\rightarrow$ trust prior more
  - Small $\sigma$ (clean) $\rightarrow$ trust observation
- Best linear estimator for Gaussian prior

## For Inverse Problems

- Score network gives $\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$
- Evaluate $A(\hat{\mathbf{x}}_0)$ for data fidelity

**Tweedie: the bridge between score and denoised reconstruction**

# Example: DPS on a Gaussian Mixture Model

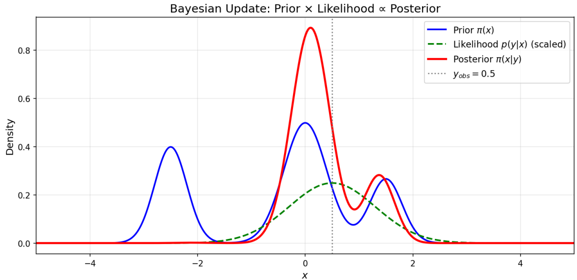
## A Tractable Example Where Everything is Analytical

**Prior**: 3-component GMM

$\pi(\mathbf{x}) = \sum_{k=1}^{3} w_k \mathcal{N}(\mathbf{x}; \mu_k, \sigma_k^2)$

- Means: $\mu = [-2.5, 0.0, 1.5]$
- Stds: $\sigma = [0.3, 0.4, 0.3]$, Weights: $w = [0.3, 0.5, 0.2]$

**Likelihood**: $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \mathbf{x}, \sigma_y^2)$, $y_{\text{obs}} = 0.5$, $\sigma_y = 0.8$




## Analytical Posterior (also a GMM!) — precision addition + evidence weighting

$$\tilde{\sigma}_k^2 = \frac{\sigma_k^2 \sigma_y^2}{\sigma_k^2 + \sigma_y^2}, \quad \tilde{\mu}_k = \tilde{\sigma}_k^2 \left( \frac{\mu_k}{\sigma_k^2} + \frac{y_{\text{obs}}}{\sigma_y^2} \right), \quad \tilde{w}_k \propto w_k \cdot \mathcal{N}(y_{\text{obs}}; \mu_k, \sigma_k^2 + \sigma_y^2)$$

**key insight: mode at $x = -2.5$ nearly eliminated ($\tilde{w}_1 \approx 0.001$)**

# Score Functions for the GMM

## Score of a Gaussian Mixture

For GMM $p(\mathbf{x}) = \sum_k w_k \varphi_k(\mathbf{x})$ where $\varphi_k = \mathcal{N}(\mathbf{x}; \mu_k, \sigma_k^2)$:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}) = \frac{\sum_k w_k \nabla_{\mathbf{x}} \varphi_k(\mathbf{x})}{\sum_k w_k \varphi_k(\mathbf{x})} = -\sum_k r_k(\mathbf{x}) \frac{\mathbf{x} - \mu_k}{\sigma_k^2}$$

where **responsibility** $r_k(\mathbf{x}) = \frac{w_k \varphi_k(\mathbf{x})}{\sum_j w_j \varphi_j(\mathbf{x})}$ is soft assignment to component $k$

## Time-Evolved Marginal at Time $t$

Under VP diffusion: $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \epsilon$, the marginal is also a GMM:

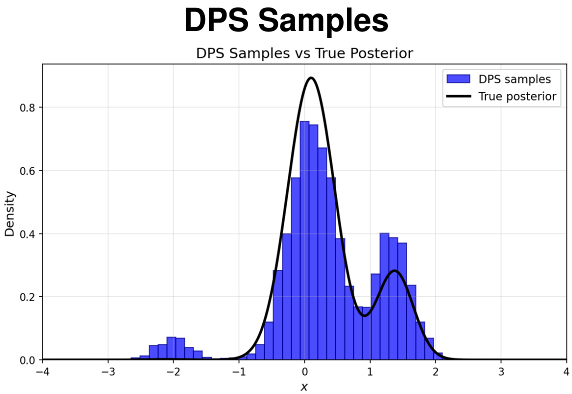
$$p_t(\mathbf{x}_t) = \sum_k w_k \mathcal{N}(\mathbf{x}_t; \alpha_t \mu_k, \alpha_t^2 \sigma_k^2 + \sigma_t^2)$$

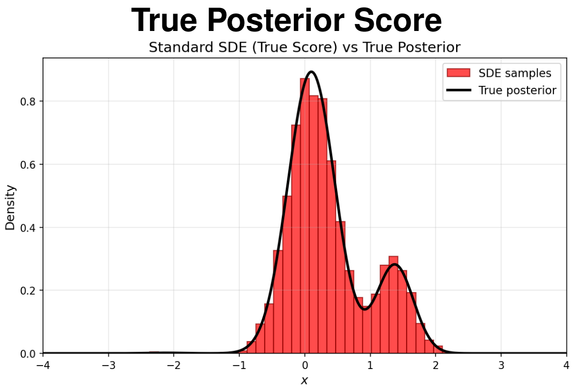
## Same Formula, Time-Evolved Parameters

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = -\sum_k r_k^{(t)}(\mathbf{x}_t) \frac{\mathbf{x}_t - \alpha_t \mu_k}{\alpha_t^2 \sigma_k^2 + \sigma_t^2}$$

**GMM score can be computed analytically. No training/approximation required!**

# Visual Comparison: DPS vs True Posterior Sampling



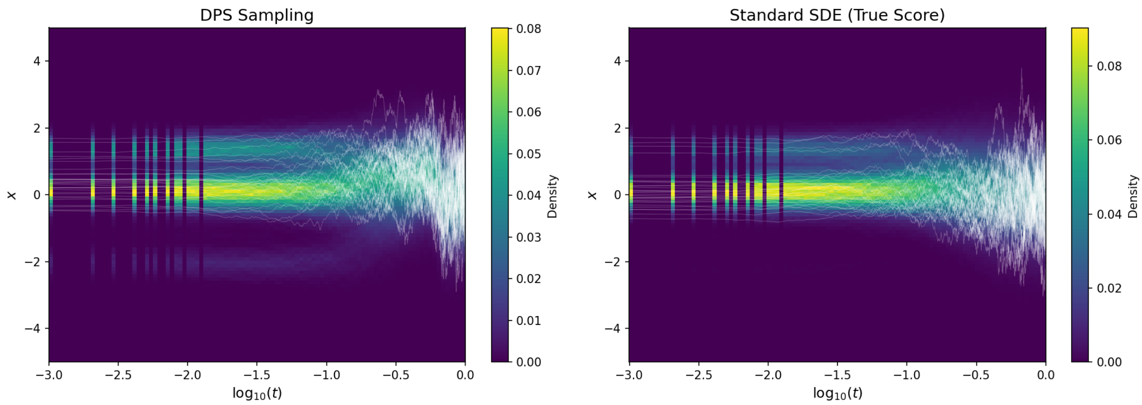




## Key Observations

- True SDE (red): matches posterior perfectly (validates implementation)
- DPS (blue): **spurious samples near** $x \approx -2$ (should be $\approx 0.1\%$!)
- DPS std = 0.82 vs true std = 0.62 (**32% inflation**)

**DPS over-represents unlikely modes due to delayed guidance**

# Spacetime Evolution: When Does DPS Fail?




**Timeline of Failure** (read right to left: $t = 1 \to t = 0$)

- **Large** $t$: Both start from $\mathcal{N}(0, 1)$—identical
- **Medium** $t$: True SDE suppresses $x \approx -2$ mode; DPS doesn't
- **Small** $t$: Too late—DPS trajectories "stuck" near wrong mode

**DPS lacks early mode discrimination $\to$ wrong final distribution**

# Why DPS Fails: The Approximation Problem $\star$

**True posterior score at time** $t$: $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$

**The problem**: We know $p(\mathbf{y}|\mathbf{x}_0)$, but need $p_t(\mathbf{y}|\mathbf{x}_t) = \int p(\mathbf{y}|\mathbf{x}_0)\, p(\mathbf{x}_0|\mathbf{x}_t)\, d\mathbf{x}_0$

**DPS Approximation**: Replace integral with **point estimate** via Tweedie

$$\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \frac{\mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}{\alpha_t} \quad \Rightarrow \quad \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \approx \frac{1}{\alpha_t} \nabla_{\hat{\mathbf{x}}_0} \log p(\mathbf{y}|\hat{\mathbf{x}}_0)$$

**Why This Fails** (recall: $\alpha_t \to 0$ as $t \to 1$, i.e., early in reverse process):

- **Multi-modal posterior**: $\hat{\mathbf{x}}_0$ may lie *between* modes $\to$ spurious $x \approx -2$ samples
- **Early guidance unstable**: $\alpha_t$ small $\to 1/\alpha_t$ explodes $\to$ must disable guidance $\to$ miss mode discrimination (spacetime plot!)

**Point estimate + delayed guidance $\to$ wrong mode weights! Better estimators needed!**

# Summary

# Σ: Inverse Problems and AI

## Ill-Posed Inverse Problems

- Direct inverse learning fails: no rescue from data/architecture
- Regularization needed: encode prior knowledge to stabilize

## Bayesian Framework

- Uncertainty is information: multi-modal posteriors reveal non-uniqueness
- Posterior enables better decisions than point estimates

## Simulation-Based Inference

- Amortized inference: pay training cost once, get fast posteriors
- Requires many training samples (forward simulations)
- COT-Flow: 30-100× reduction for expensive PDE simulators

## Diffusion Posteriors

- Not amortized: runs full diffusion process per sample
- Score additivity enables modular composition (prior + likelihood)
- Challenge: Good approximation of score of likelihood needed

# Σ: Other Open Challenges

**Fast Diffusion Sampling**

- Goal: 5-10 steps instead of 1000 (consistency models, distillation)
- Enables real-time clinical applications

**Robustness and Model Misspecification**

- What happens when prior/likelihood are wrong?
- Domain shift, out-of-distribution, uncertainty about uncertainty

**Physics-Informed Generative Models**

- Combine PDE constraints with learned priors
- Hard constraints vs soft penalties; conservation laws

# References I


- Antun, Vegard et al. (2020). “On instabilities of deep learning in image reconstruction and the potential costs of AI”. In: *Proceedings of the National Academy of Sciences* 117.48, pp. 30088–30095. DOI: 10.1073/pnas.1907377117.
- Chung, H. et al. (2023). “Diffusion Posterior Sampling for General Noisy Inverse Problems”. In: *International Conference on Learning Representations (ICLR)*.
- Cranmer, K. et al. (2020). “The Frontier of Simulation-Based Inference”. In: *Proceedings of the National Academy of Sciences* 117.48, pp. 30055–30062.
- Efron, Bradley (2011). “Tweedie’s Formula and Selection Bias”. In: *Journal of the American Statistical Association* 106.496, pp. 1602–1614. DOI: 10.1198/jasa.2011.tm11181.
- Greenberg, David S. et al. (2019). “Automatic Posterior Transformation for Likelihood-Free Inference”. In: *International Conference on Machine Learning (ICML)*, pp. 2404–2414.
- Kawar, B. et al. (2022). “Denoising Diffusion Restoration Models”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35, pp. 23593–23606.

# References II


- Papamakarios, G. et al. (2019). "Sequential Neural Likelihood: Fast Likelihood-Free Inference with Autoregressive Flows". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 837–848.
- Wang, Z. O. et al. (2023). "Efficient Neural Network Approaches for Conditional Optimal Transport with Applications in Bayesian Inference". In: *arXiv preprint arXiv:2310.16975*.