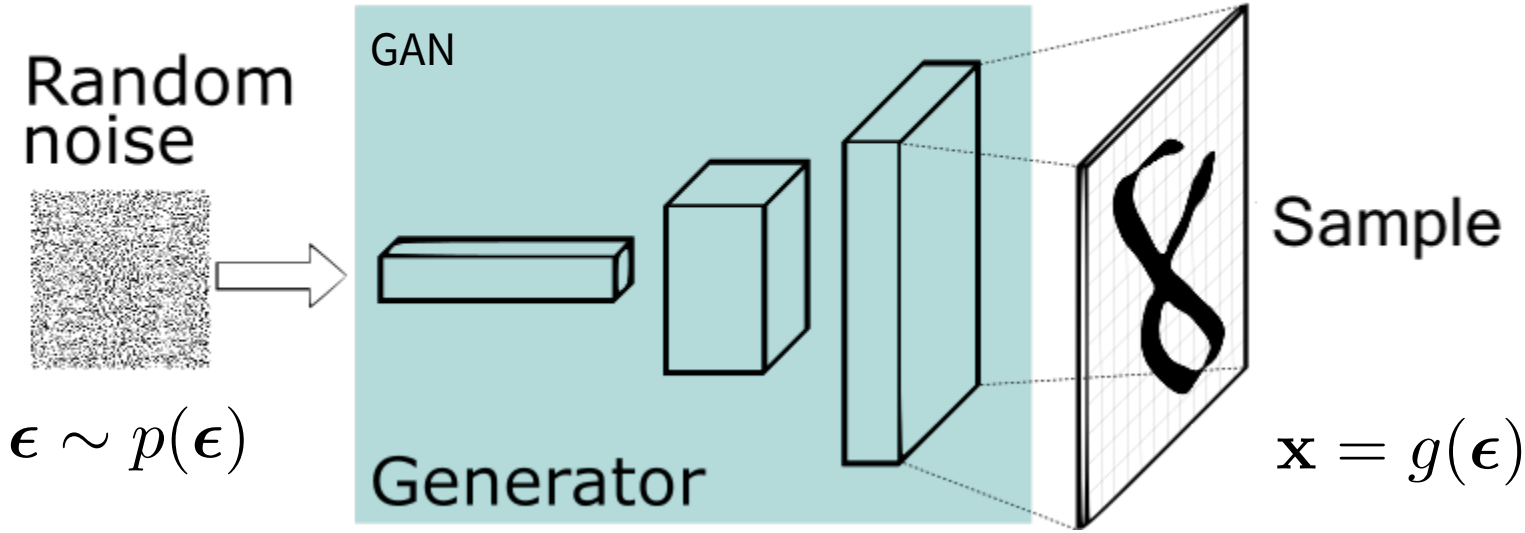
Generative Modeling by Estimating Gradients of the Data Distribution

YANG SONG
STEFANO ERMON

Stanford AI Lab

Representations of Probability Distributions

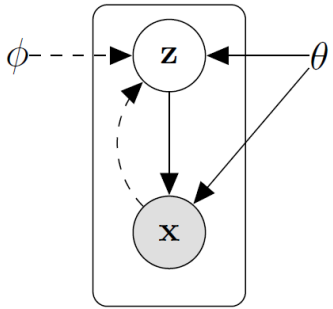
Implicit models: directly represent the sampling process



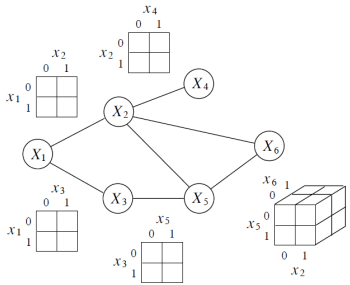
Cons: hard to train, no likelihood, no principled model comparisons

Representations of Probability Distributions

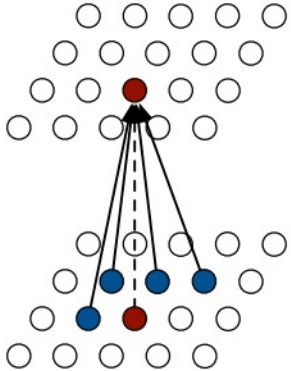
Explicit models: represent a probability density/mass function $p(\mathbf{x})$



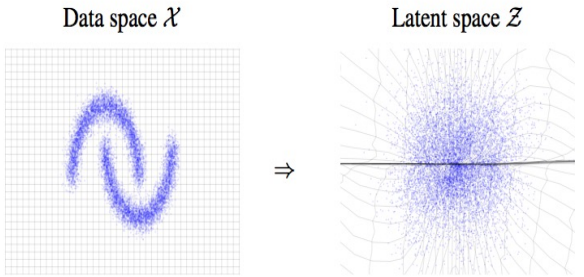
Bayesian networks (e.g., VAEs)



MRF



Autoregressive models



Flow models

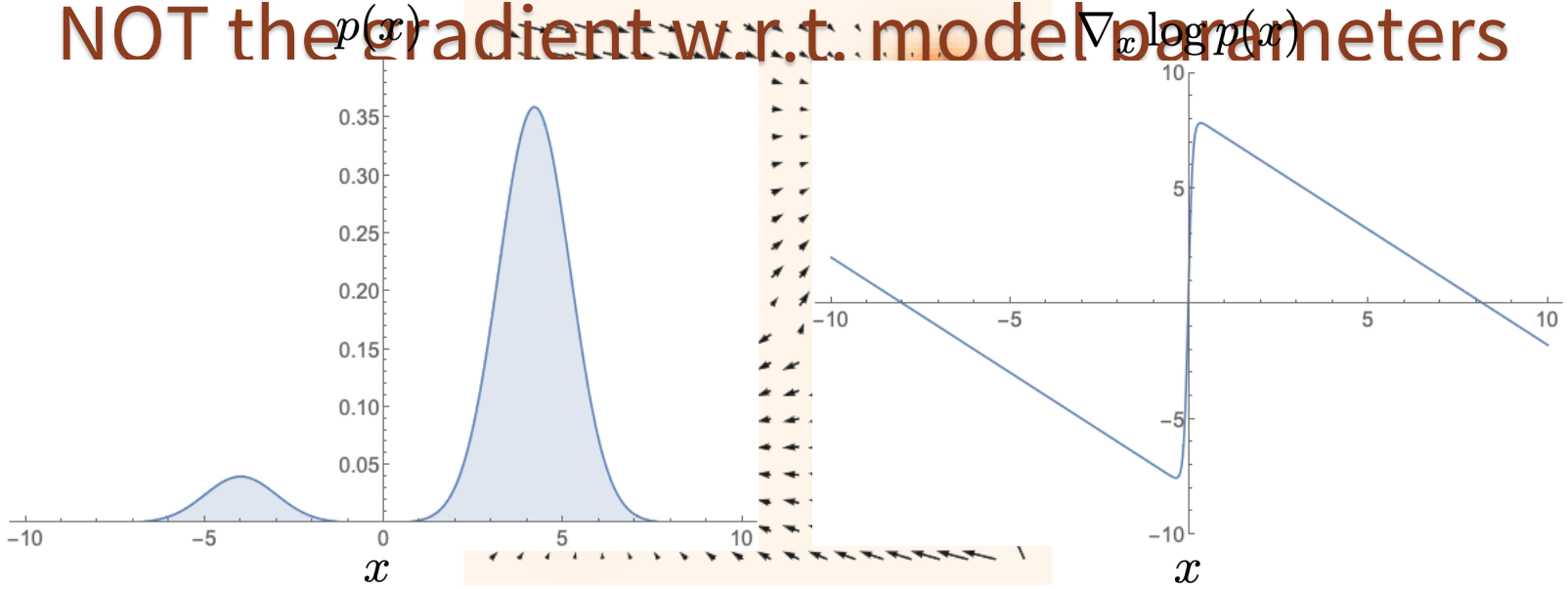
Cons: need to be normalized → balance expressivity and tractability

Representation of Probability Distributions

This talk: The gradient of a probability density w.r.t. the input dimensions

$$\nabla_{\mathbf{x}} \log p(\mathbf{x})$$ Score

NOT the gradient w.r.t. model parameters



(pdf and score)

Score Estimation

- **Given:** i.i.d. samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p_{\text{data}}(\mathbf{x})$
- **Task:** Estimating the score $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$
- **Score Model:** A trainable vector-valued function $s_\theta(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}^D$
- **Objective:** How to compare two vector fields of scores?

$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - s_\theta(\mathbf{x})\|_2^2]$$

$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ (Fisher divergence)

- Integration by parts

$$\mathbb{E}_{p_{\text{data}}}\left[\frac{1}{2}\|s_\theta(\mathbf{x})\|_2^2 + \text{trace}(\nabla_{\mathbf{x}} s_\theta(\mathbf{x}))\right]$$

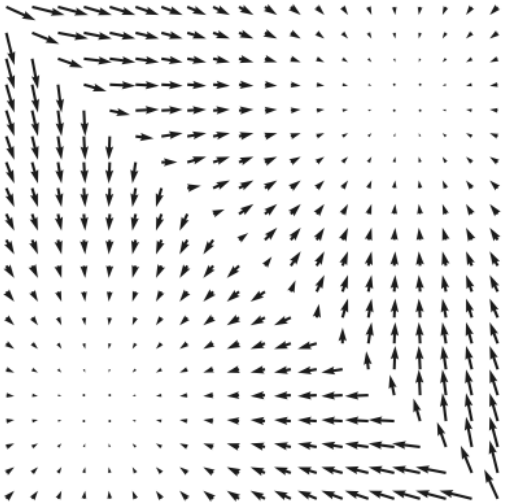
$$\approx \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}\|s_\theta(\mathbf{x}_i)\|_2^2 + \text{trace}(\nabla_{\mathbf{x}} s_\theta(\mathbf{x}_i))\right]$$

Average Euclidean distance

Score Matching

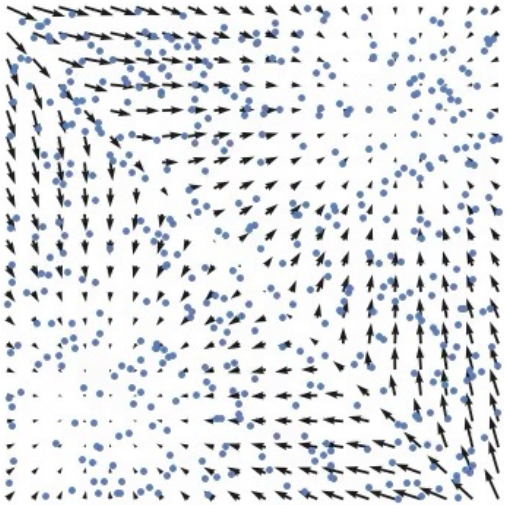
Hyvärinen (2005)

From Scores to Samples: Langevin Dynamics



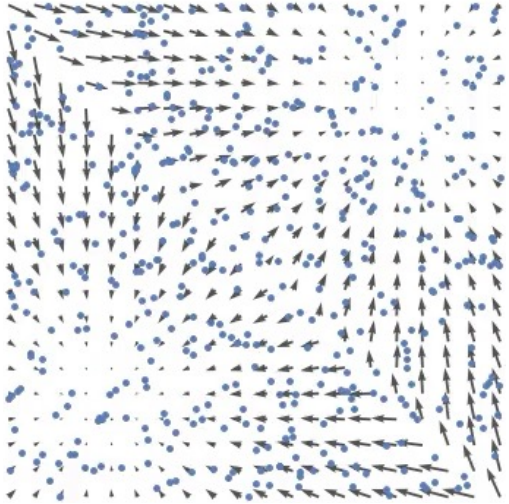
Scores

$$s_\theta(\mathbf{x})$$



Follow the scores

$$\tilde{\mathbf{x}}_{t+1} \leftarrow \tilde{\mathbf{x}}_t + \frac{\epsilon}{2} s_\theta(\tilde{\mathbf{x}}_t)$$

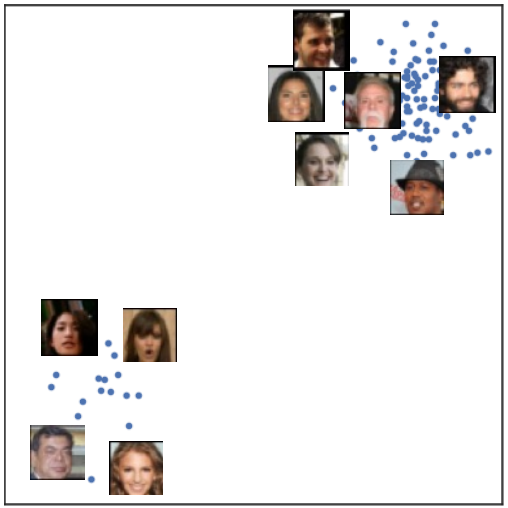


Follow noisy scores:
Langevin dynamics

$$\mathbf{z}_t \sim \mathcal{N}(0, I)$$

$$\tilde{\mathbf{x}}_{t+1} \leftarrow \tilde{\mathbf{x}}_t + \frac{\epsilon}{2} s_\theta(\tilde{\mathbf{x}}_t) + \sqrt{\epsilon}\mathbf{z}_t$$

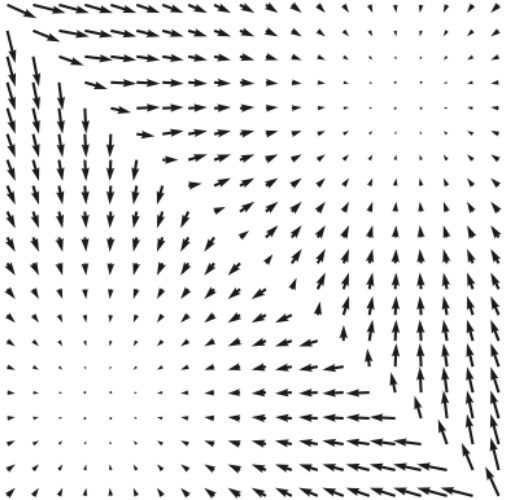
Score-Based Generative Modeling



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p_{\text{data}}(\mathbf{x})$$

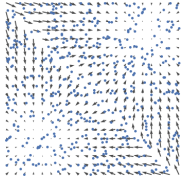
Score Matching



Scores

$$s_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$

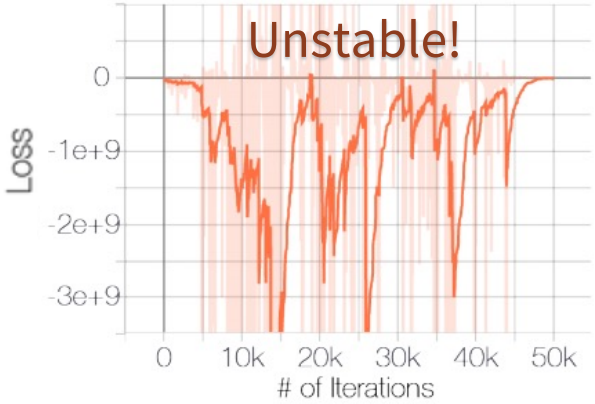
Langevin dynamics



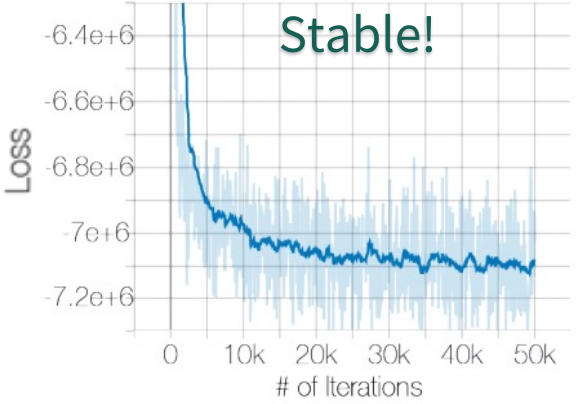
New samples

Adding Noise to Data for Well-Defined Scores

- Scores can be undefined when
  - The support of data distribution is on a low-dimensional manifold
  - The data distribution is discrete
- Solution: adding noise

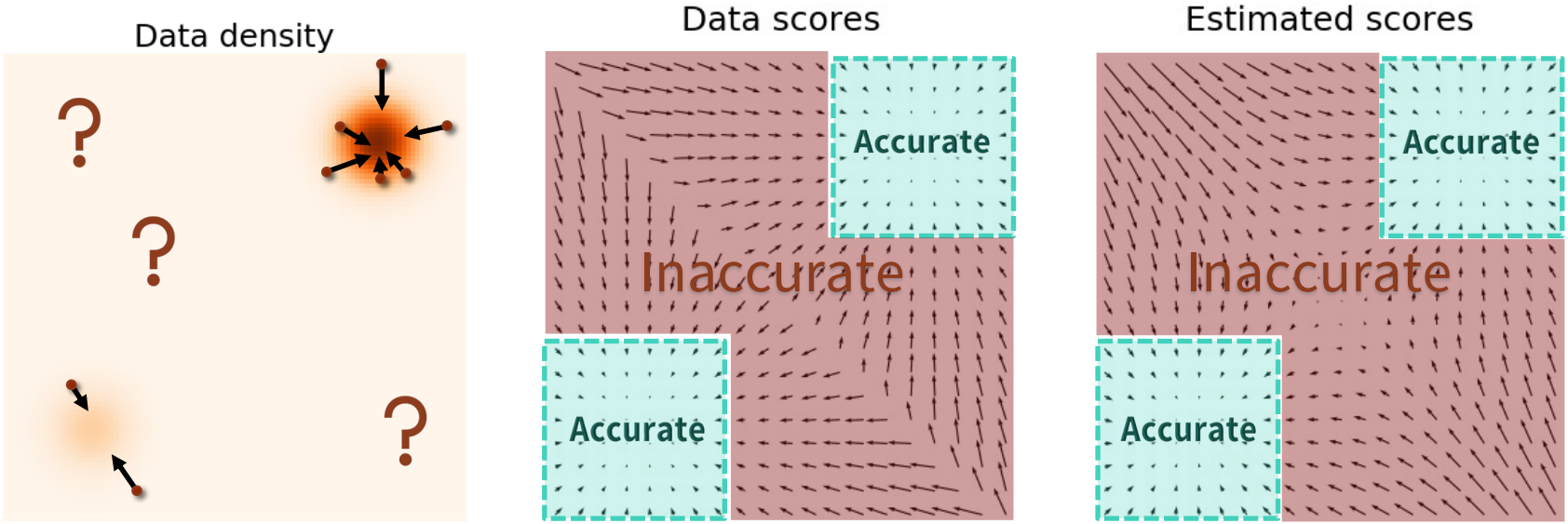


Data unperturbed



Data perturbed with $\mathcal{N}(0; 0.0001)$

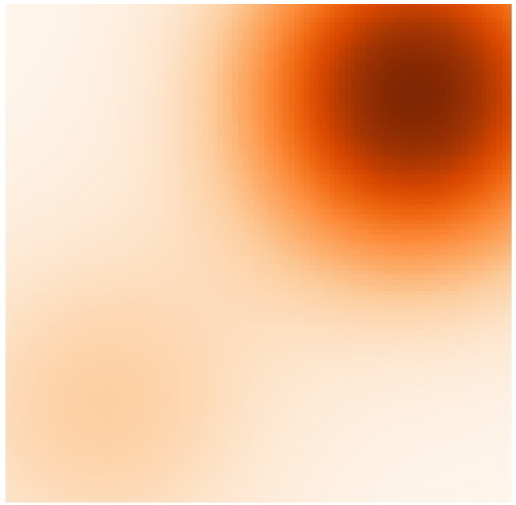
Challenge in Low Data Density Regions

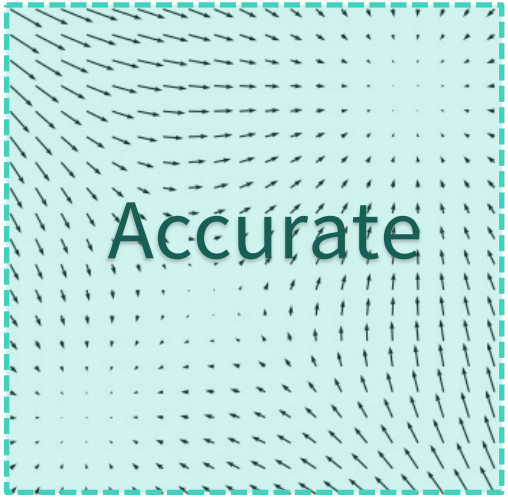


$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[\|\nabla_{\mathbf{x}}\log p_{\text{data}}(\mathbf{x}) - s_\theta(\mathbf{x})\|_2^2] \approx \frac{1}{2N}\sum_{i=1}^{N}\|\nabla_{\mathbf{x}}\log p_{\text{data}}(\mathbf{x}_i) - s_\theta(\mathbf{x}_i)\|_2^2$$

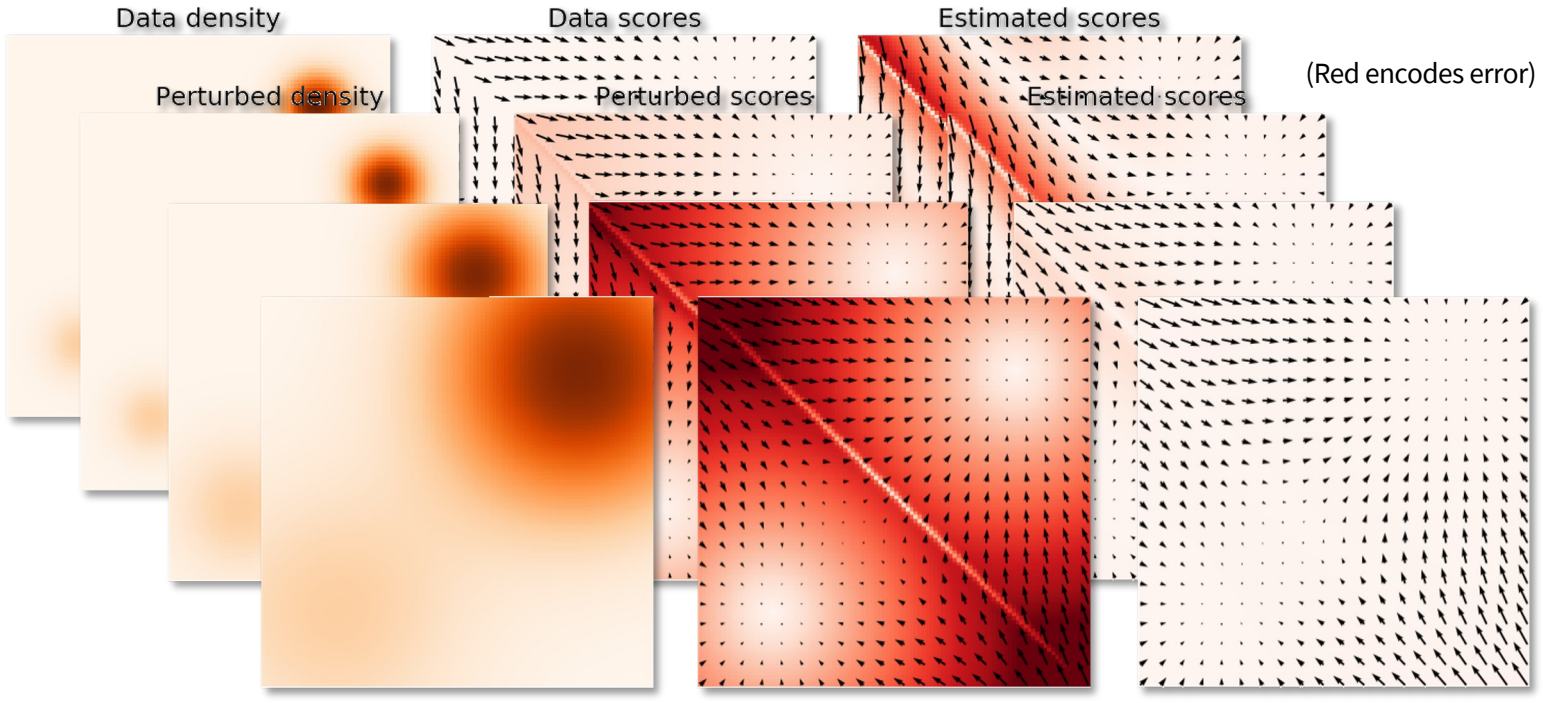
Adding Noise to Data for Better Score Estimation

- Random noise provides samples in low data density regions.

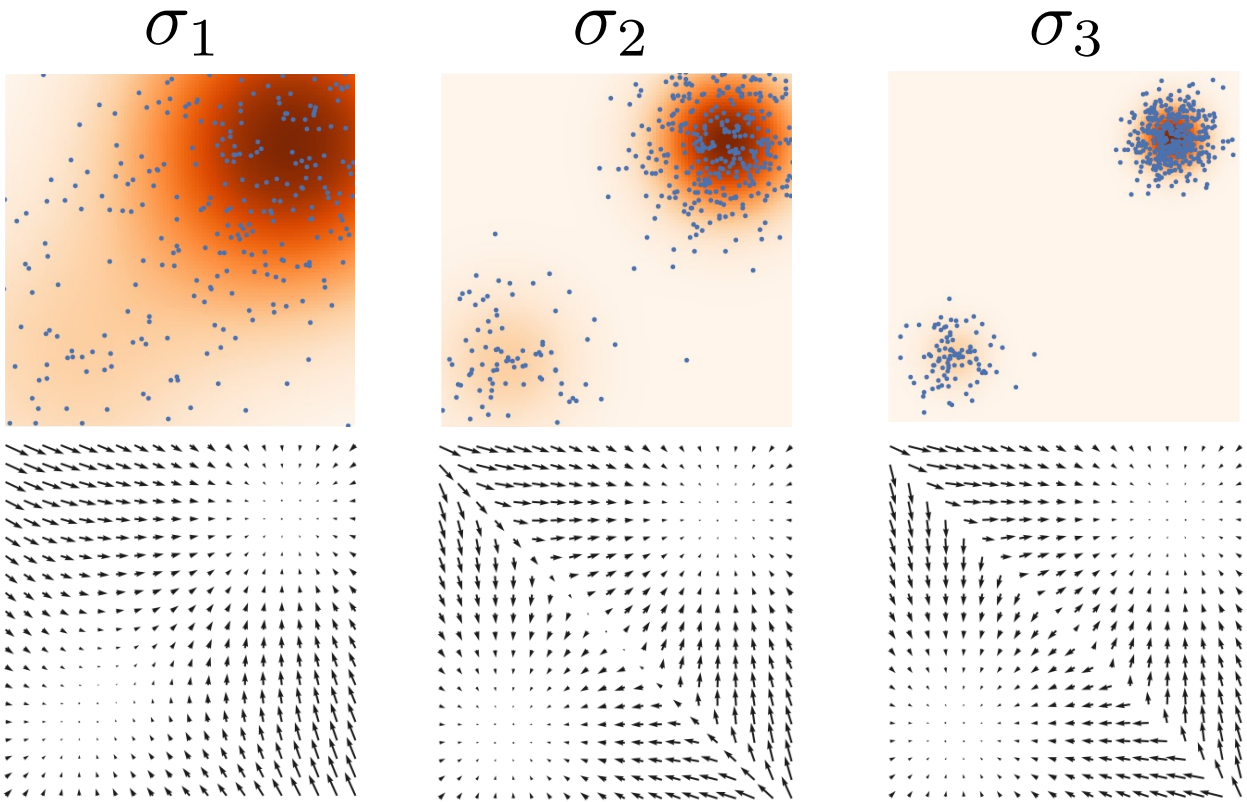


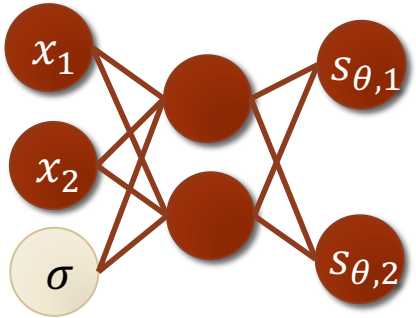


Trading off Sample Quality and Estimation Accuracy



Joint Score Estimation via Noise Conditional Score Networks



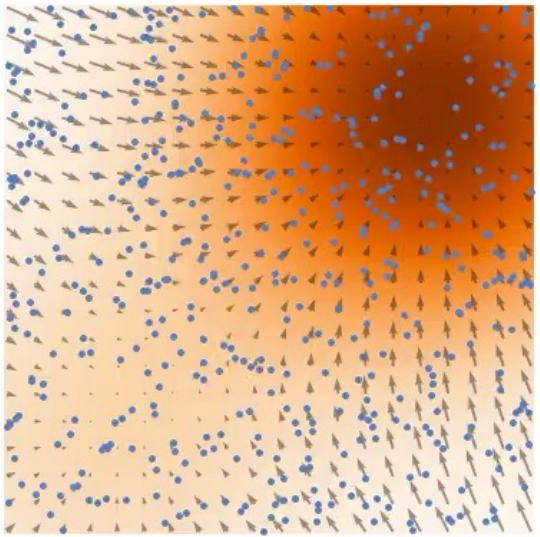


Noise Conditional Score Network (NCSN)

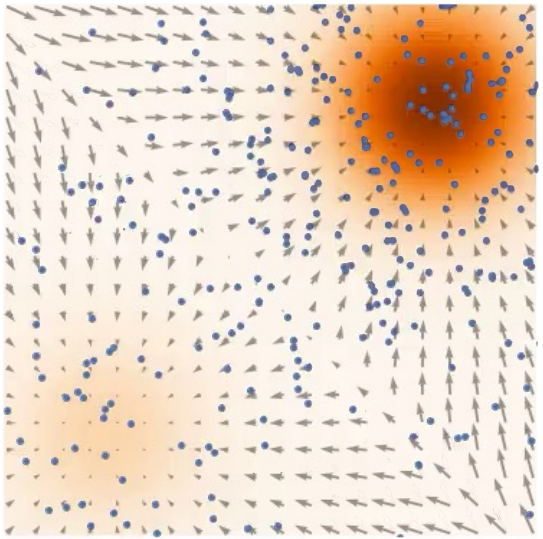
Stanford University

Annealed Langevin Dynamics: Joint Scores to Samples

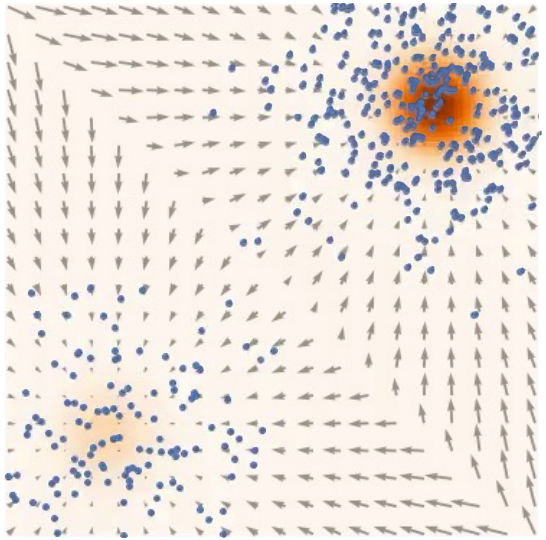
- Sample using $\sigma_1, \sigma_2, \cdots, \sigma_L$ sequentially with Langevin dynamics.
- Anneal down the noise level.
- Samples used as initialization for the next level.



σ_1

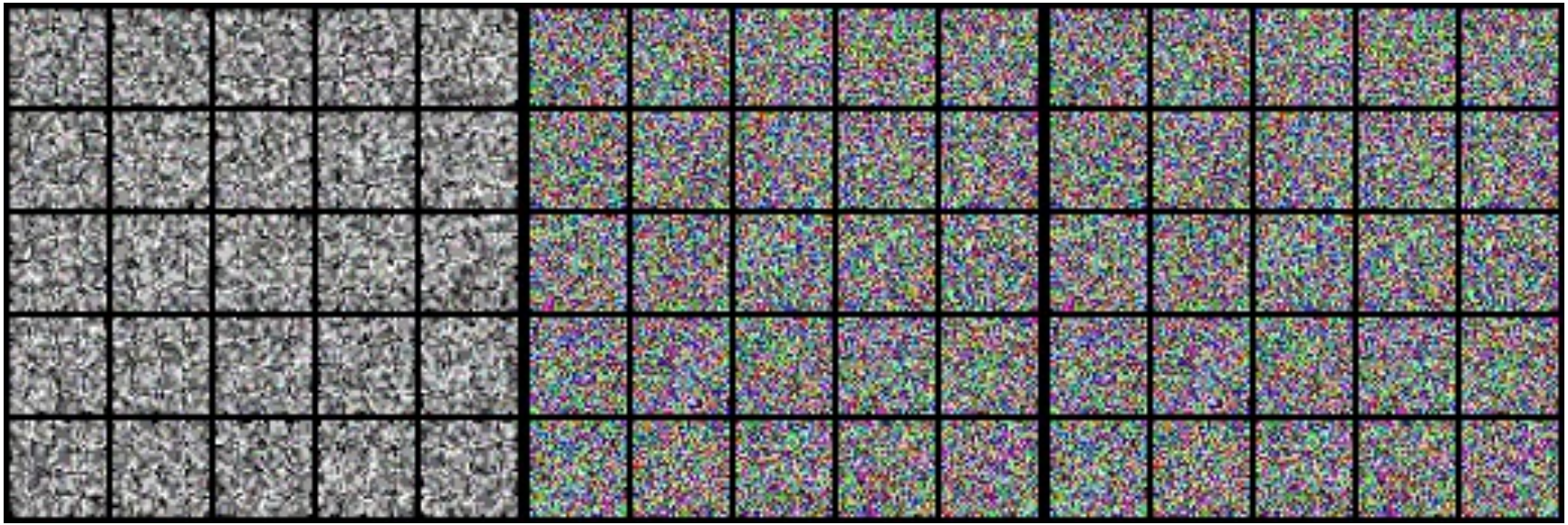


σ_2



σ_3

Experiments: Sampling



Experiments: Sample Quality

CIFAR-10 Unconditional

Model	Inception Score (higher is better)	FID score (lower is better)
PixelCNN	4.60	65.93
EBM	6.02	40.58
SNGAN	8.22 ± 0.05	21.7
ProgressiveGAN	8.80 ± 0.05	-
NCSN (ours)	**8.87 ± 0.12**	**25.32**

Experiments: Inpainting

Conclusion

- Score-based generative modeling
  - **No need to be normalized / invertible**
    - Flexible architecture choices
  - **No minimax optimization**
    - stable training
    - a natural measurement of training progress / model comparison

- **Adding noise** and **annealing the noise levels** are critical

- Better or comparable sample quality to GANs.

Related Work

- Generative Stochastic Networks (Bengio et al. (2013), Alain et al. (2016))
  - Sampling starts **close to data points.**
  - Need **MCMC during training** with walkback.

- Nonequilibrium Thermodynamics (Sohl-Dickstein et al. (2015)), Infusion Training (Bordes et al. (2017)), Variational Walkback (Goyal et al. (2017))
  - **Likelihood**-based training.
  - Need **MCMC during training.**

Experiments: Nearest Neighbors

Future Directions

- How to apply score-based generative modeling to discrete data?
- Theoretical guidance on how to choose noise levels?
- Better architecture for higher resolution image generation?
- Improved score estimation?