How Good Are Deep Generative Models for Solving Inverse Problems?

Shichong Peng, Alireza Moazeni, Ke Li
APEX Lab, School of Computing Science, Simon Fraser University

NEURAL INFORMATION PROCESSING SYSTEMS

Abstract

Deep generative models have shown impressive capability in tackling inverse problems. However, the **validity** of the model-generated solutions w.r.t. the forward problem and the reliability of associated **uncertainty estimates** remain understudied.

In this work, we evaluate recent **diffusion-based, GAN-based, IMLE-based** methods on three challenging inverse problems. We find that the IMLE-based **CHIMLE**[1] method outperforms other methods in terms of producing valid solutions and reliable uncertainty estimates.

Background

Diffusion Models

Diffusion models consider a forward problem that progressively adds Gaussian noise to the data according to a predefined variance schedule, known as the *forward process*:

$$q(x_{1:T}|x_0) \coloneqq \prod_{t=1}^{T} q(x_t|x_{t-1}), q(x_t|x_{t-1}) \coloneqq \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I})$$

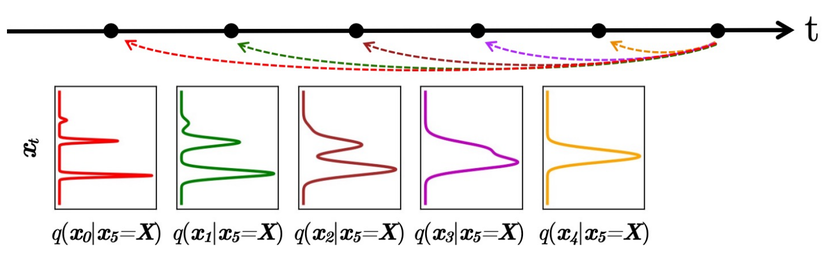
Original Data — Variance Schedule

Diffusion models aim to invert the forward process with another Markov chain with Gaussian transition kernels, known as the *reverse process*, starting at $p(x_T) \coloneqq \mathcal{N}(x_T; 0, \mathrm{I})$:

$$p_\theta(x_{0:T}) \coloneqq p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), p_\theta(x_{t-1}|x_t) \coloneqq \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Predicted Mean/Covariance

However, the reverse process can only truly invert the forward process if each time step is **infinitesimal**; otherwise, the transition kernel in the **true** reverse process is **not** Gaussian[4].



In practice, **a finite T** is used, so the Gaussian transition kernel assumption introduces approximation errors. Therefore, the assumptions of diffusion models are not met when the forward problem is not that of **adding Gaussian noise** or when the number of time steps is **small**.

Generative Adversarial Networks (GANs)

GANs consist of a generator G_θ and a discriminator D_ϕ and optimize the following objective:

$$\min_\theta \max_\phi \mathbb{E}_{y \sim \hat{p}_{data}}[\log D_\phi(y)] + \mathbb{E}_{z \sim \mathcal{N}(0,I)}[\log(1 - D_\phi(G_\theta(z)))]$$

However, the model may get stuck at a local optimum where the generator only captures a subset of the modes in the empirical data distribution. This is also known as mode collapse.

Implicit Maximum Likelihood Estimation (IMLE)

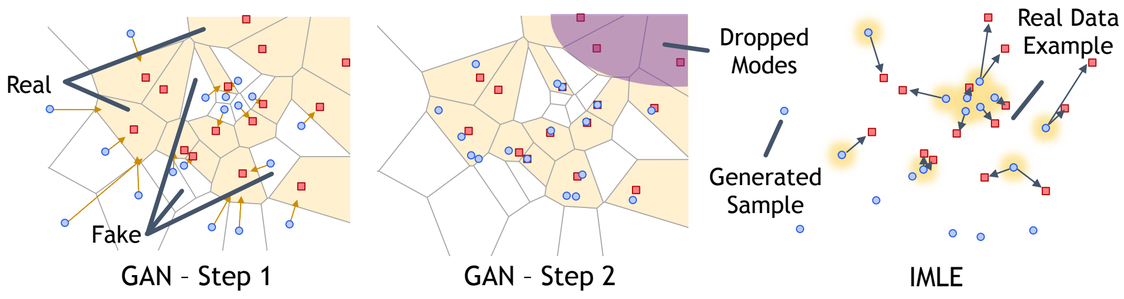
IMLE[2] uses a generator G_θ like GANs, but it does not use a discriminator nor adversarial training. IMLE generates a pool of m samples and pulls the closest sample to each real data example y. The objective function takes the form:

$$\min_\theta \mathbb{E}_{z_1,\dots,z_m \sim \mathcal{N}(0,I)} \left[\sum_{i=1}^{n} \min_{j \in \{1,\dots,m\}} d(G_\theta(z_j), y_i) \right]$$

A sample from the generator

Pool of latent codes — Select the closest sample — Real data example

Intuitively, IMLE overcomes **mode collapse** by ensuring ***each*** real data example has ***some*** similar generated samples. On the contrary, GANs only ensure ***each*** generated sample is similar to ***some*** real data examples, which is the opposite.



Conditional Hierarchical IMLE (CHIMLE)

Conditional IMLE (cIMLE)[3] extends IMLE to the conditional setting by introducing a conditioning input x for each observed image y (real data example). The objective function is as follows:

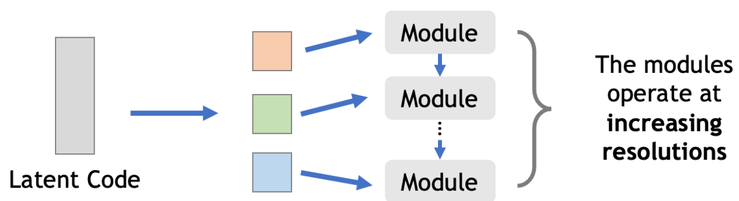
$$\min_\theta \mathbb{E}_{z_{1,1},\dots,z_{n,m} \sim \mathcal{N}(0,I)} \left[\sum_{i=1}^{n} \min_{j \in \{1,\dots,m\}} d(G_\theta(x_i, z_{i,j}), y_i) \right]$$

Pool Size — Conditioning Input

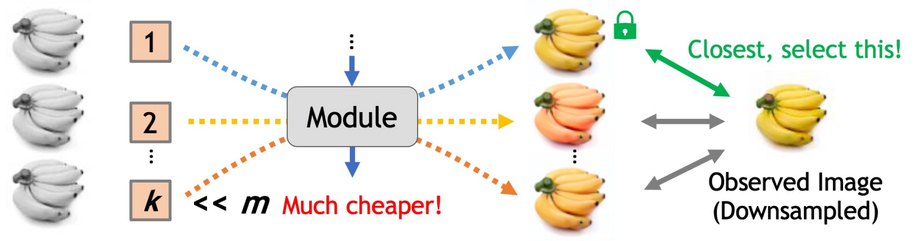
To generate a "good" sample which is close to the observed image, cIMLE **requires a large pool size m**. However, sampling is expensive, so only a limited number of samples can be generated in practice, and this limits the selected sample quality.

A recent method, CHIMLE[1], overcomes this limitation by introducing a hierarchical sampling algorithm that **efficiently searches for** a "good" sample as if it was selected from a large pool of samples.

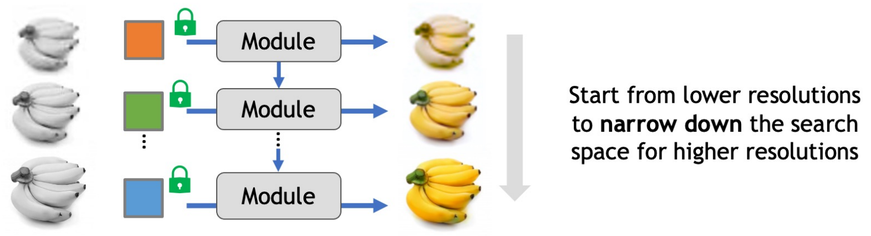
Step 1: Divide latent code into components, each operating at a different resolution.



Step 2: Select the value for each code component that produces the intermediate output closest to the observed image (downsampled to the same resolution).



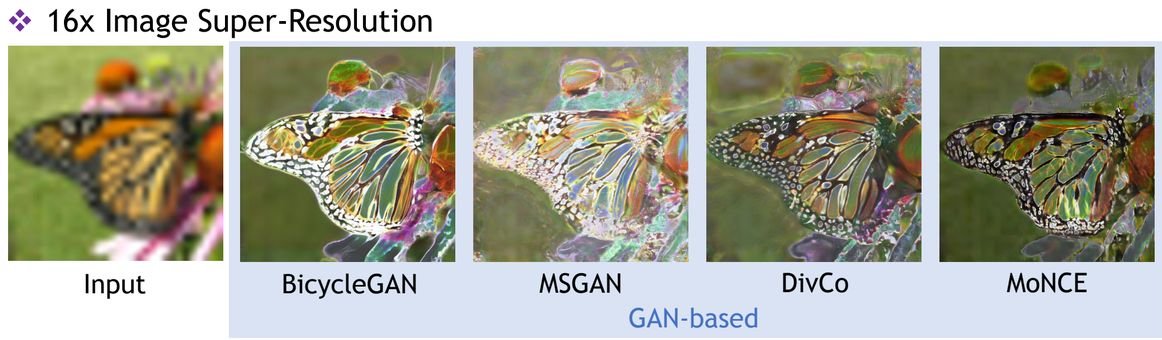
Step 3: Construct the full latent code from the code component starting at the lowest resolution and fix the selected value for lower resolutions before moving on to a higher resolution.



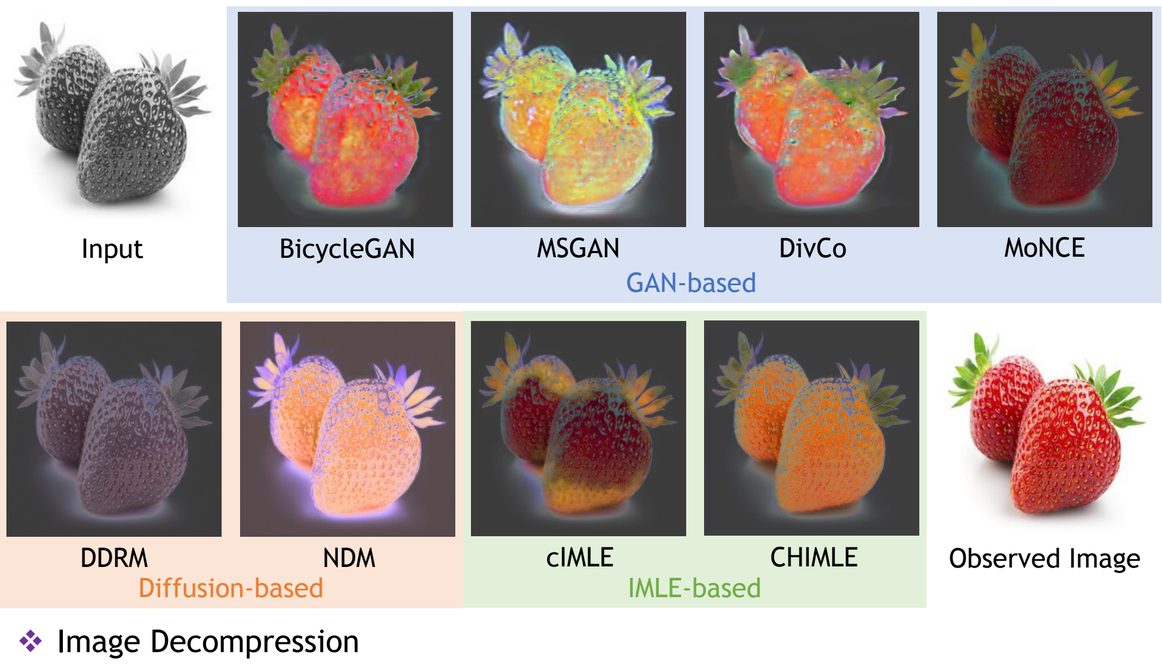
Uncertainty Quantification

We measure the model uncertainty using a sampling-based conformal prediction method from [5]. The constructed confidence intervals are shown below:

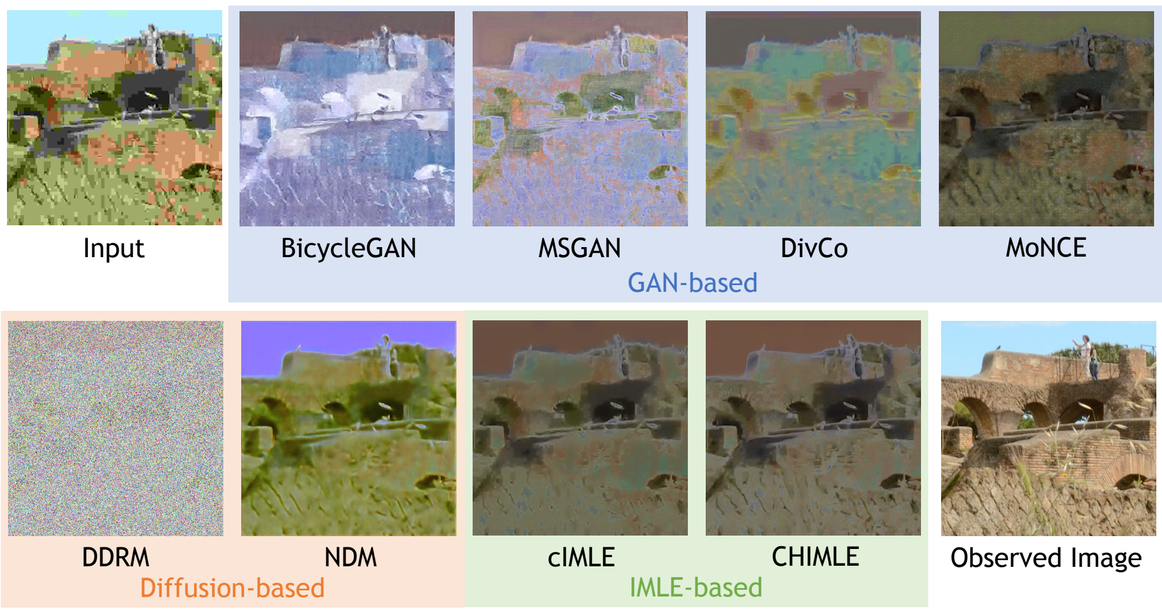
❖ 16x Image Super-Resolution



❖ Colourization



❖ Image Decompression



Output Validity

We evaluate the output validity of each method by comparing the original input to the solution to the forward problem applied to the generated image.

	Super-Resolution (SR)			Colourization (Col)			Image Decompression (DC)		
	LPIPS ↓	*PSNR* ↑	*SSIM* ↑	*LPIPS* ↓	*PSNR* ↑	*SSIM* ↑	*LPIPS* ↓	*PSNR* ↑	*SSIM* ↑
GAN-based:									
BicyleGAN	0.105	22.31	0.832	0.322	20.28	0.716	0.349	19.75	0.781
MSGAN	0.135	20.83	0.810	0.370	18.98	0.665	0.416	17.24	0.712
DivCo	0.136	20.60	0.763	0.336	18.84	0.687	0.356	18.94	0.731
MoNCE	0.091	27.41	0.943	0.030	39.52	**0.990**	0.213	25.78	0.854
Diffusion-based:									
DDRM	0.045	29.57	0.948	0.127	28.21	0.922	0.539	19.07	0.459
NDM	0.082	23.50	0.854	0.057	37.34	0.951	0.525	14.15	0.575
IMLE-based:									
cIMLE	0.040	28.57	0.949	0.022	36.12	0.981	0.261	22.39	0.790
CHIMLE	**0.009**	**33.26**	**0.988**	**0.011**	**41.25**	**0.990**	**0.191**	**26.24**	**0.877**

References

[1] Peng, S., Moazeni, A., & Li, K. (2022). CHIMLE: Conditional Hierarchical IMLE for Multimodal Conditional Image Syntehsis. *NeurIPS*.

[2] Li, K., & Malik, J. (2018). Implicit Maximum Likelihood Estimation. *ArXiv, abs/1809.09087*.

[3] Li, K., Peng, S., Zhang, T., & Malik, J. (2020). Multimodal Image Synthesis with Conditional Implicit Maximum Likelihood Estimation. *IJCV*, 1-22.

[4] Xiao, Z., Kreis, K., & Vahdat, A. (2022). Tackling the Generative Learning Trilemma with Denoising Diffusion GANs. *ICLR*.

[5] Horwitz, E., & Hoshen, Y. (2022). Conffusion: Confidence Intervals for Diffusion Models. *ArXiv, abs/2211.09795*.