# Safe and Informative Computational Imaging via Conformal Prediction and Generative Models

**Phil Schniter**

(Joint work with Jeffrey Wen, Matt Bendel, and Rizwan Ahmad)

The Ohio State University

Supported in part by the NIH under grant R01-EB029957

University of Texas at Austin
December 2024

# Imaging inverse problems

Imaging inverse problems:

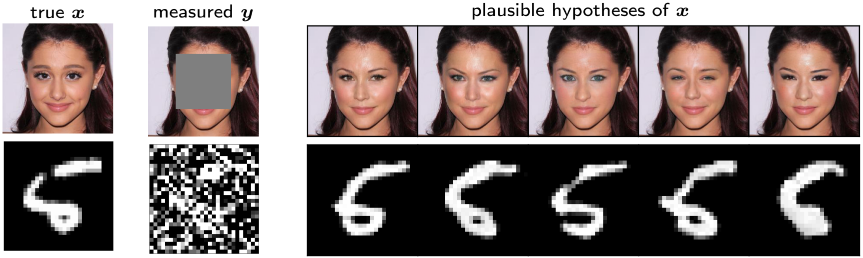
- Unknown image $\boldsymbol{x}$
- Measurements $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x})$
    - $\mathcal{M}(\cdot)$ masks, distorts, and/or corrupts $\boldsymbol{x}$ with noise.
    - Examples: denoising, deblurring, inpainting, super-resolution, phase retrieval, computed tomography (CT), magnetic resonance imaging (MRI), etc.
- Typical goals
    - Recover image $\boldsymbol{x}$
    - Extract quantitative information from $\boldsymbol{x}$ (e.g., probability of a pathology)

Challenges:

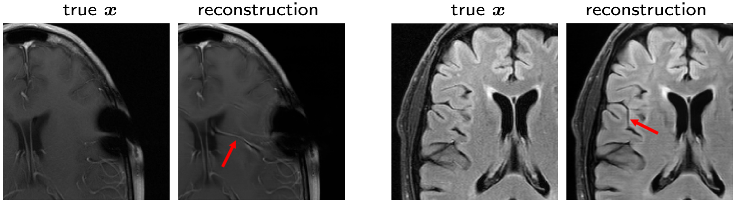
- The inverse problem is often ill-posed: many hypotheses of $\boldsymbol{x}$ can explain the measurements $\boldsymbol{y}$
- Modern methods can hallucinate:
    - Produce nice-looking $\widehat{\boldsymbol{x}}$ that differ from true $\boldsymbol{x}$ in important ways

# Examples of ill-posedness and hallucinations

Fundamental ill-posedness:


true $\boldsymbol{x}$
measured $\boldsymbol{y}$
plausible hypotheses of $\boldsymbol{x}$

Undesired hallucinations[1]:



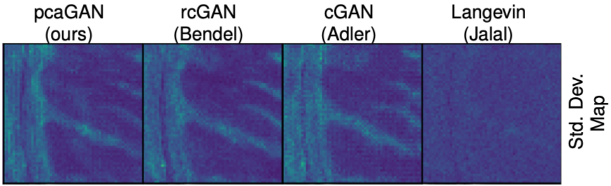
[1] Muckley et al'21

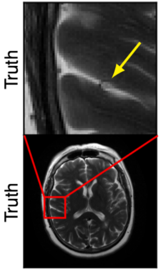
# Uncertainty quantification

- We'd like to quantify the uncertainty or error in the image recovery $\widehat{\boldsymbol{x}}$
  - If the error was large, we could collect new/more measurements

- Especially important in safety-critical applications (e.g., medical imaging)

- Most uncertainty quantification methods produce pixel-wise uncertainty maps (e.g., posterior variance) that look like





- But how useful are these uncertainty maps in practice?

# Outline

1. Probabilistic bounds on recovered-image accuracy
2. Probabilistic bounds for quantitative/task-based imaging
3. Efficient posterior sampling via diffusion with fast iterative renoising
4. Fast posterior sampling with regularized cGANs

# Probabilistic bounds on recovered-image accuracy

- Say we have an image-recovery method $\boldsymbol{r}(\cdot)$ that produces $\widehat{\boldsymbol{x}}_0 = \boldsymbol{r}(\boldsymbol{y}_0)$
  - Subscript $_0$ indicates "test" quantities, not calibration, as we'll soon see ...

- We'd like to know the "accuracy" of $\widehat{\boldsymbol{x}}_0$ relative to the true $\boldsymbol{x}_0$

- To measure accuracy, we'll use an arbitrary image-quality metric $\boxed{z_0 = m(\widehat{\boldsymbol{x}}_0, \boldsymbol{x}_0)}$ like
  - PSNR, SSIM[1] ... higher preferred (we'll focus on this below)
  - LPIPS[2], DISTS[3] ... lower preferred
  - In any case, $z_0$ is unknown in practice

- Is it possible to guarantee the accuracy of $\widehat{\boldsymbol{x}}_0$, i.e., construct a bound $\beta_0(\boldsymbol{y}_0)$ such that

$$\Pr\{Z_0 \geq \beta_0(\boldsymbol{Y}_0)\} \geq 1 - \alpha$$

  for some chosen error rate $\alpha$? (Here, capital letters denote random variables)

[1] Wang et al'04, [2] Zhang et al'18, [3] Ding et al'20

# An impractical bound

- Say we have a perfect posterior sampler generating $c$ image samples $\{\widetilde{\boldsymbol{x}}_0^{(j)}\}_{j=1}^c$. Then

$$\{\boldsymbol{x}_0, \widetilde{\boldsymbol{x}}_0^{(1)}, \dots, \widetilde{\boldsymbol{x}}_0^{(c)}\} \overset{iid}{\sim} p_{\boldsymbol{X}_0 \,|\, \boldsymbol{Y}_0}(\cdot \,|\, \boldsymbol{y}_0)$$

- Define the corresponding image-accuracy samples $\widetilde{z}_0^{(j)} \triangleq m(\widehat{\boldsymbol{x}}_0, \widetilde{\boldsymbol{x}}_0^{(j)})$. Then

$$\{z_0, \widetilde{z}_0^{(1)}, \dots, \widetilde{z}_0^{(c)}\} \overset{iid}{\sim} p_{Z_0 \,|\, \boldsymbol{Y}_0}(\cdot \,|\, \boldsymbol{y}_0)$$

- Thus an accuracy lower bound $\beta_0$ that holds with probability $1-\alpha$, i.e.,

$$\Pr\{Z_0 \geq \beta_0 \,|\, \boldsymbol{Y}_0 = \boldsymbol{y}_0\} = 1 - \alpha$$

  can be constructed using an infinite number of perfect posterior samples:

$$\beta_0 = \lim_{c \to \infty} \widehat{\beta}_0 \quad \text{with} \quad \widehat{\beta}_0 \triangleq \operatorname{EmpQuant}(\alpha, \{\widetilde{z}_0^{(j)}\}_{j=1}^c)$$

  - Why? Assuming the random variable $Z_0 \,|\, \boldsymbol{Y}_0 = \boldsymbol{y}_0$ is continuous, we recognize $\beta_0$ as its $\alpha$th quantile. Then $\{\widetilde{z}_0^{(j)}\}_{j=1}^c \xrightarrow{d} Z_0 \,|\, \boldsymbol{Y}_0 = \boldsymbol{y}_0$ implies $\widehat{\beta}_0 \to \beta_0$

# A conformal bound

- In practice, we have only a finite number $c$ of imperfect posterior samples
- We propose to design a valid lower bound using conformal prediction[1]
    - Assume we have $n$ calibration samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ in addition to the test measurements $\boldsymbol{y}_0$
    - Construct approximate $c$-sample bounds as before:

$$\widehat{\beta}_i = \mathrm{EmpQuant}(\alpha, \{\widetilde{z}_i^{(j)}\}_{j=1}^c) \text{ for } i = 0, \dots, n$$

- 
    - Also compute "bound-violation scores" $s_i \triangleq \widehat{\beta}_i - z_i$ for $i = 1, \dots, n$ (i.e., $s_i > 0$ means bound violated)
    - Using the set $d_{\text{cal}} \triangleq \{s_i\}_{i=1}^n$ of calibration scores, compute a bound correction term

$$\widehat{\lambda}(d_{\text{cal}}) = \mathrm{EmpQuant}\left(\tfrac{\lceil (1-\alpha)(n+1) \rceil}{n}, \{s_i\}_{i=1}^n\right)$$

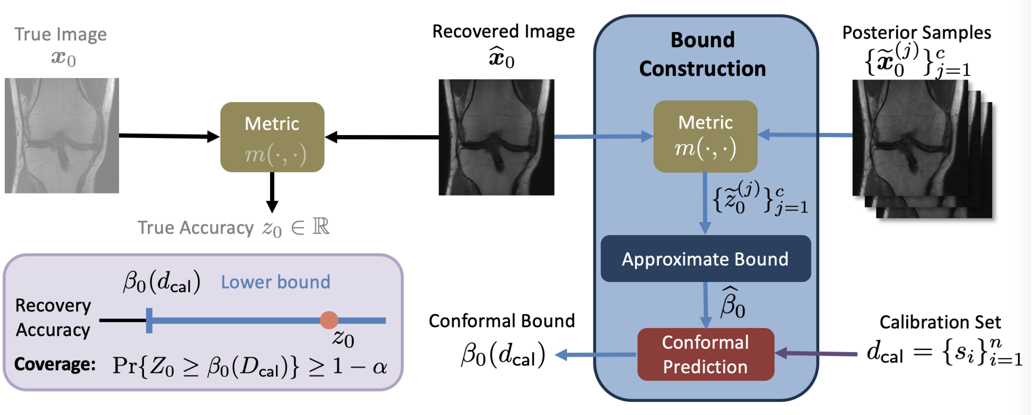
- 
    - Form the final "test" lower bound as $\beta_0(\boldsymbol{y}_0, d_{\text{cal}}) = \widehat{\beta}_0 - \widehat{\lambda}(d_{\text{cal}})$
- This conformal bound obeys[2] the marginal coverage guarantee

$$1 - \alpha \le \Pr\{Z_0 \ge \beta_0(\boldsymbol{Y}_0, D_{\text{cal}})\} \le 1 - \alpha + \tfrac{1}{n+1}$$

assuming that $\{S_0, S_1, \dots, S_n\}$ are statistically exchangeable

[1] Vovk,Gammerman,Shafer'05, [2] Lei,G'Sell,Rinaldo,Tibshirani,Wasserman'18,

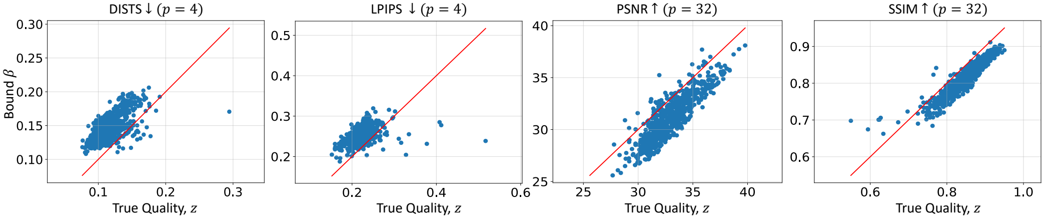
# Summary of lower-bound procedure


True Image
$x_0$
Metric
$m(\cdot,\cdot)$
True Accuracy $z_0 \in \mathbb{R}$
Recovered Image
$\widehat{x}_0$
Bound Construction
Metric
$m(\cdot,\cdot)$
$\{\widetilde{z}_0^{(j)}\}_{j=1}^c$
Approximate Bound
$\widehat{\beta}_0$
Conformal Prediction
Posterior Samples
$\{\widetilde{x}_0^{(j)}\}_{j=1}^c$
Calibration Set
$d_{\text{cal}} = \{s_i\}_{i=1}^n$
Conformal Bound
$\beta_0(d_{\text{cal}})$
$\beta_0(d_{\text{cal}})$
Lower bound
Recovery Accuracy
$z_0$
Coverage: $\Pr\{Z_0 \geq \beta_0(D_{\text{cal}})\} \geq 1-\alpha$

- Similar techniques can be used to construct upper bounds on lower-preferred image accuracy metrics like LPIPS and DISTS

# Example: Bounding recovery accuracy in MRI

- Scatter plots of $(z_0, \beta_0)$ from fastMRI knee recovery @ acceleration $R = 8$ using a conditional normalizing flow[1]:



The red line indicates where the bound would be exact

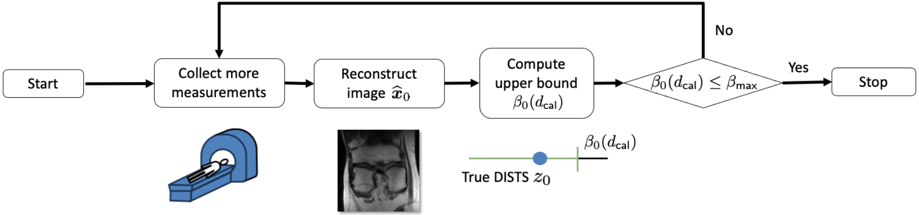
- Validation of marginal coverage using $10\,000$ Monte-Carlo trials (each with a random $70\%$ test / $30\%$ calibration split):

| target coverage $1 - \alpha$ | average empirical coverage |
|---|---|
| $0.95$ | $0.9504 \pm 0.0001$ |

[1] Wen,Ahmad,S'23

# Application: Multi-round MRI acquisition

- Consider acquiring over multiple rounds (i.e., $R \in \{16, 8, 4, 2, 1\}$), stopping as soon at the conformal upper bound on DISTS[1] is good enough (i.e., $\leq \beta_{\text{max}}$)



Start

Collect more measurements

Reconstruct image $\widehat{\boldsymbol{x}}_0$

Compute upper bound $\beta_0(d_{\text{cal}})$

$\beta_0(d_{\text{cal}}) \leq \beta_{\text{max}}$

No

Yes

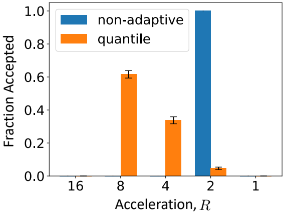
Stop

$\beta_0(d_{\text{cal}})$

True DISTS $z_0$

- Multi-round acquisition achieves much higher average acceleration than single-round acquisition:

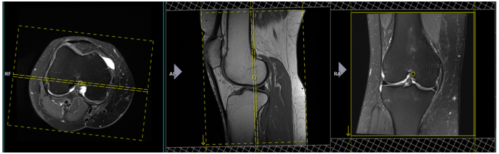
| method | avg acceleration | empirical coverage |
|---|---|---|
| single-round | $2.000 \pm 0.0000$ | $0.9505 \pm 0.0001$ |
| multi-round | $5.422 \pm 0.0001$ | $0.9461 \pm 0.0001$ |

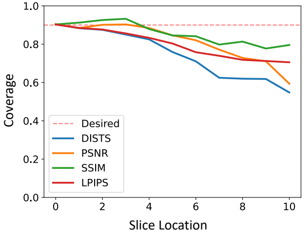


[1] Kastryulin, Zakirov, Pezzotti, Dylov'23

# Effect of distribution shift

- Recall that the marginal coverage guarantee $\Pr\{Z_0 \geq \beta_0(\boldsymbol{Y}_0, D_{\textsf{cal}})\} \geq 1-\alpha$ requires exchangeability of the test and calibration scores $\{S_0, S_1, \ldots, S_n\}$
- This can fail if the distribution shifts between calibration and test
- We investigate this phenomenon by calibrating on only the center slices of fastMRI knee volumes and testing on a variety of slices, from the center to the edge





  - PSNR and SSIM are more robust than LPIPS and DISTS
- Recent theoretical results[1,2] show that it's possible to extend coverage to a TV ball around the calibration distribution by choosing a more conservative error-rate $\alpha$

[1] Cauchois,Gupta,Ali,Duchi'24, [2] Oliveira,Orenstein,Ramos,Romano'24

# Outline

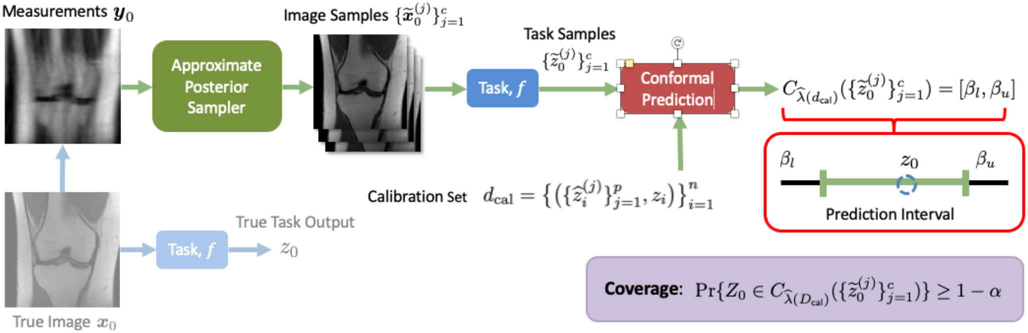
1. Probabilistic bounds on recovered-image accuracy
2. Probabilistic bounds for quantitative/task-based imaging
3. Efficient posterior sampling via diffusion with fast iterative renoising
4. Fast posterior sampling with regularized cGANs

# Quantitative / task-based imaging

- Again consider measurements $\boldsymbol{y}_0 = \mathcal{A}(\boldsymbol{x}_0)$ and some recovery $\widehat{\boldsymbol{x}}_0 = r(\boldsymbol{y}_0)$
- But now say that our goal is to extract quantitative information about $\boldsymbol{x}_0$
- Example: Does the MRI knee image $\boldsymbol{x}_0$ indicate a meniscus tear?
  - Say we've trained & calibrated a soft-output binary classifier $f(\cdot)$ on clean images
  - Naively applying $f(\cdot)$ to imperfect recoveries $\widehat{\boldsymbol{x}}_0$ would give unreliable results
  - Instead, we want to estimate $f(\boldsymbol{x}_0)$ from $\boldsymbol{y}_0$ (without knowing $\boldsymbol{x}_0$)
- More generally, one may wish to estimate a generic $\boxed{z_0 = f(\boldsymbol{x}_0) \in \mathbb{R}}$ given $\boldsymbol{y}_0$
- Can one construct guaranteed upper and lower bounds on $z_0$?

# Conformal prediction of the true task output

- Our approach is similar to before, in that we combine posterior image sampling with conformal prediction
- But instead of a one-sided bound, we construct a prediction interval $C_\lambda = [\beta_l, \beta_u]$ that is guaranteed to contain the true task output $z_0$ with probability $1-\alpha$



Measurements $\boldsymbol{y}_0$
Approximate Posterior Sampler
Image Samples $\{\tilde{\boldsymbol{x}}_0^{(j)}\}_{j=1}^c$
Task Samples
$\{\tilde{z}_0^{(j)}\}_{j=1}^c$
Task, $f$
Conformal Prediction
$C_{\hat{\lambda}(d_{\text{cal}})}(\{\tilde{z}_0^{(j)}\}_{j=1}^c) = [\beta_l, \beta_u]$
$\beta_l$
$z_0$
$\beta_u$
Prediction Interval
Calibration Set $d_{\text{cal}} = \{(\{\tilde{z}_i^{(j)}\}_{j=1}^p, z_i)\}_{i=1}^n$
True Task Output
Task, $f$
$z_0$
True Image $\boldsymbol{x}_0$

**Coverage:** $\Pr\{Z_0 \in C_{\hat{\lambda}(D_{\text{cal}})}(\{\tilde{z}_0^{(j)}\}_{j=1}^c)\} \geq 1-\alpha$

# Constructing the prediction set

Several options for constructing the prediction set, e.g.,

- Absolute residual (AR):
$$C_\lambda(\{\widetilde{z}^{(j)}\}_{j=1}^c) = [\overline{z}-\lambda, \overline{z}+\lambda], \quad \overline{z} = \tfrac{1}{c}\textstyle\sum_{j=1}^c \widetilde{z}^{(j)}$$
- Locally weighted residual (LWR):
$$C_\lambda(\{\widetilde{z}^{(j)}\}_{j=1}^c) = [\overline{z}-\sigma_z\lambda, \overline{z}+\sigma_z\lambda], \quad \sigma_z^2 = \tfrac{1}{c}\textstyle\sum_{j=1}^c (\widetilde{z}^{(j)}-\overline{z})^2$$
- Conformalized quantile regression[1] (CQR):
$$C_\lambda(\{\widetilde{z}^{(j)}\}_{j=1}^c) = \big[\,\mathrm{EmpQuant}(\tfrac{\alpha}{2}, \{\widetilde{z}^{(j)}\}_{j=1}^c) - \lambda, \mathrm{EmpQuant}(1-\tfrac{\alpha}{2}, \{\widetilde{z}^{(j)}\}_{j=1}^c) + \lambda\big]$$

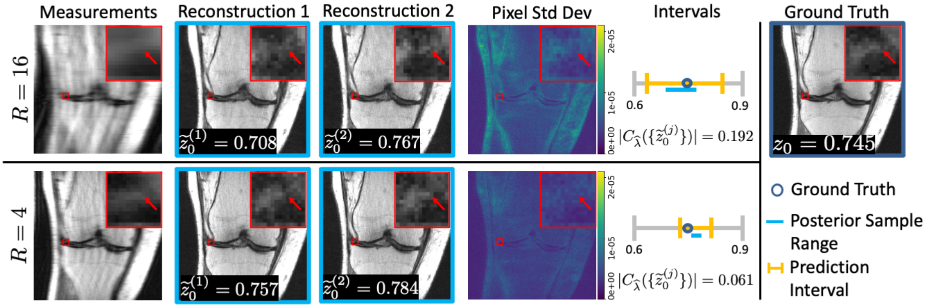
Adaptivity:

- The interval length $|C_\lambda|$ provides a quantitative measure of uncertainty
- The AR method has a fixed $|C_\lambda|$, while LWR and CQR adapt the length $|C_\lambda|$ to $\boldsymbol{y}_0$

[1] Romano,Patterson,Candès'19

# Example: Predicting meniscus-tears in knee MRI

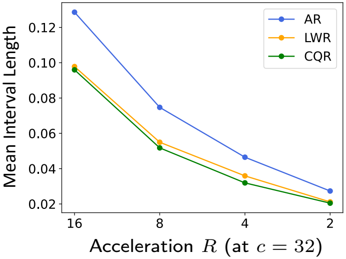
- We trained and calibrated a ResNet50 to output meniscus-tear probability $z_0 = f(\boldsymbol{x}_0)$ using clean images from fastMRI+
- From $R$-accelerated measurements $\boldsymbol{y}_0$, we compute a prediction interval $C_\lambda$ that contains the true $z_0$ with 99% probability
- The conformal bound uses $c$ posterior samples from a conditional normalizing flow[1]

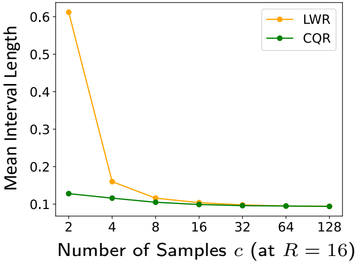

Measurements
Reconstruction 1
Reconstruction 2
Pixel Std Dev
Intervals
Ground Truth
$R = 16$
$\tilde{z}_0^{(1)} = 0.708$
$\tilde{z}_0^{(2)} = 0.767$
2e-05
1e-05
0e+00
0.6
0.9
$|C_{\hat{\lambda}}(\{\tilde{z}_0^{(j)}\})| = 0.192$
$z_0 = 0.745$
$R = 4$
$\tilde{z}_0^{(1)} = 0.757$
$\tilde{z}_0^{(2)} = 0.784$
2e-05
1e-05
0e+00
0.6
0.9
$|C_{\hat{\lambda}}(\{\tilde{z}_0^{(j)}\})| = 0.061$
○ Ground Truth
— Posterior Sample Range
⊢⊣ Prediction Interval

[1] Wen,Ahmad,S'23

# Effect of prediction-set method

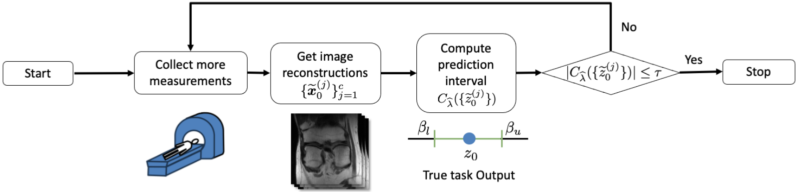
- Recall that the interval length $|C_\lambda|$ acts as a quantitative measure of task uncertainty
- Below we plot $|C_\lambda|$ vs acceleration $R$ and # posterior samples $c$ for the AR, LWR, and CQR bounds
- In this application, CQR yields the least uncertainty and is robust to the number of posterior samples





# Application: Multi-round MRI acquisition

- Consider acquiring over multiple rounds (i.e., $R \in \{16, 8, 4, 2, 1\}$), stopping as soon at the task uncertainty is small enough ($|C_\lambda| \le \tau$ for $\tau = 0.1$)



Start

Collect more measurements

Get image reconstructions $\{\tilde{\boldsymbol{x}}_0^{(j)}\}_{j=1}^{c}$

Compute prediction interval $C_{\hat{\lambda}}(\{\tilde{z}_0^{(j)}\})$

$|C_{\hat{\lambda}}(\{\tilde{z}_0^{(j)}\})| \le \tau$

No

Yes

Stop

$\beta_l$

$\beta_u$

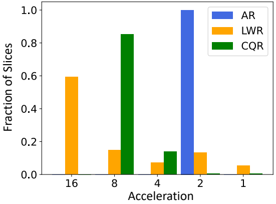
$z_0$

True task Output

- The adaptive bounding schemes achieve much higher average acceleration rates than the non-adaptive AR scheme:

| Method | Average Acceleration | Empirical Coverage | Average Center Error @ $R = 2$ |
|---|---|---|---|
| AR | $2.000$ | $0.991 \pm 0.008$ | $0.032 \pm 0.017$ |
| LWR | $5.157$ | $0.992 \pm 0.005$ | $0.020 \pm 0.002$ |
| CQR | $6.762$ | $0.987 \pm 0.008$ | $0.044 \pm 0.009$ |

Center Error $\triangleq |z_0 - \frac{\beta_l + \beta_u}{2}|$



# Outline

1 Probabilistic bounds on recovered-image accuracy

2 Probabilistic bounds for quantitative/task-based imaging

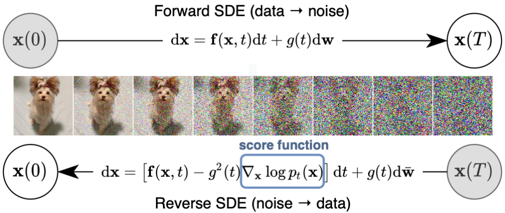
3 Efficient posterior sampling via diffusion with fast iterative renoising

4 Fast posterior sampling with regularized cGANs

# Diffusion methods

- Diffusion methods are powerful ways to sample from a complex distribution $p(\boldsymbol{x})$

- The forward process gradually adds noise to $\boldsymbol{x}(0) \sim p(\boldsymbol{x})$. The reverse process starts with pure noise $\boldsymbol{x}(T)$ and gradually denoises, eventually generating a sample from $p(\boldsymbol{x})$


Forward SDE (data → noise)
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$
$\mathbf{x}(T)$
score function
$\mathbf{x}(0)$
$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$
$\mathbf{x}(T)$
Reverse SDE (noise → data)

- To discretize, we'll assume VP DDPM, where step $t \in \{1, \ldots, T\}$ provides

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

- Although the reverse process is usually written using the score function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)$, it can also be written using the MMSE denoiser $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t\}$ via Tweedie's rule

$$\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t) = \frac{\sqrt{\alpha_t}\,\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t\} - \boldsymbol{x}_t}{1 - \alpha_t}$$

# Solving inverse problems with diffusion

- In inverse problems, we're given noisy/masked/distorted measurements $\boldsymbol{y} = \mathcal{A}(\boldsymbol{x}_0)$, from which we aim to recover $\boldsymbol{x}_0$

- As we saw earlier, there's value in sampling from the posterior distribution $p(\boldsymbol{x}_0|\boldsymbol{y})$ rather than just constructing a point estimate of $\boldsymbol{x}_0$

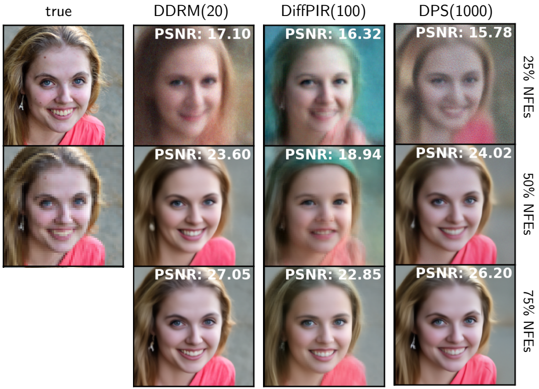
- Diffusion can be configured for posterior sampling by using $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$ in place of $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)$, or equivalently by using $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}\}$ in place of $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t\}$

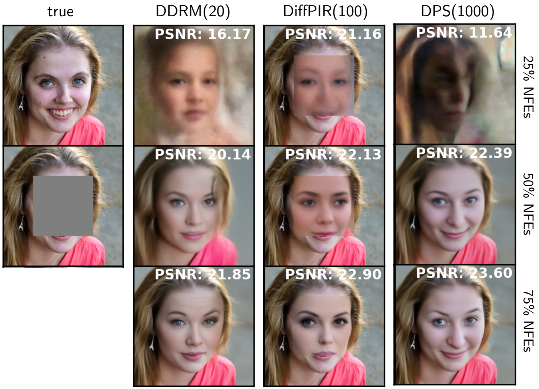
- Many strategies have been proposed to approximate $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t|\boldsymbol{y})$ or $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}\}$ using some combination of a pretrained denoiser/score-fxn and the likelihood function $p(\boldsymbol{y}|\boldsymbol{x}_0)$
  - Popular methods include DDRM[1], DPS[2], DDNM[3], $\Pi$GDM[4], DiffPIR[5], DDS[6], etc.

[1] Kawar et al'22, [2] Chung et al'23, [3] Wang et al'23, , [4] Song et al'23, , [5] Zhu et al'23, , [6] Chung et al'24

# The main challenge

- How can one best approximate $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{y}\}$ under computational constraints?
- Current methods can be evaluated by visualizing and computing the MSE of their $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{y}\}$ approximations, since the exact $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{y}\}$ minimizes MSE

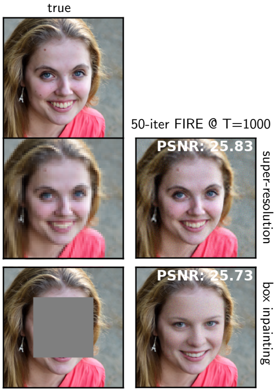




# Fast Iterative REnoising

- We propose an iterative approximation of $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t,\boldsymbol{y}\}$ that we call Fast Iterative REnoising (FIRE)

- For linear inverse problems $\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x}_0+\boldsymbol{w}$ with $\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\sigma_w^2\boldsymbol{I})$, FIRE iterates the following steps after initializing $\boldsymbol{r}=\boldsymbol{x}_t$ and $\sigma_r^2=(1-\alpha_t)/\alpha_t$:
  1. $\widetilde{\boldsymbol{x}}_0 \leftarrow \text{Denoise}(\boldsymbol{r};\sigma_r^2), \quad \sigma_x^2 \leftarrow \mathbb{E}\|\widetilde{\boldsymbol{x}}_0-\boldsymbol{x}_0\|^2/d$
  2. $\widehat{\boldsymbol{x}}_0 \leftarrow \arg\min_{\boldsymbol{x}} \frac{1}{\sigma_w^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|^2+\frac{1}{\sigma_x^2}\|\widetilde{\boldsymbol{x}}_0-\boldsymbol{x}\|^2$
  3. $\sigma_r^2 \leftarrow \sigma_r^2/\rho$ for some $\rho>1$ ... decrease renoising variance
  4. **renoise**: set $\boxed{\boldsymbol{r}\leftarrow\widehat{\boldsymbol{x}}_0+\boldsymbol{n}}$ with colored Gaussian $\boldsymbol{n}$ that gives $\boldsymbol{r}\sim\mathcal{N}(\boldsymbol{x}_0,\sigma_r^2\boldsymbol{I})$

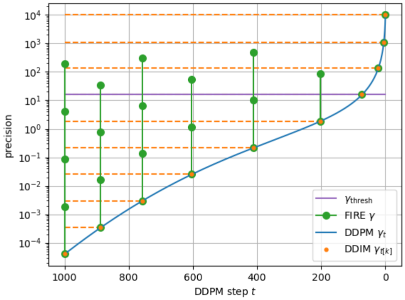


- Key idea: renoising ensures that the denoiser sees AWGN, consistent with how it was trained!

- Two options for renoising: exact SVD-based, or approximate $(\boldsymbol{A},\boldsymbol{A}^{\mathsf{T}})$-based

# DDfire: Putting the FIRE into diffusion

- FIRE can be plugged into any diffusion reverse process
- But, because FIRE uses multiple NFEs per $\mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{x}_t, \boldsymbol{y}\}$, we subsample the diffusion steps $\{t\}$ using DDIM and schedule the FIRE iterations to meet a given total-NFE budget
- We allocate FIRE iterations using a "waterfilling" procedure, which is best illustrated using *inverse* variances, i.e., precisions:
- Basically, waterfilling ensures that FIRE's final-iteration denoiser-input-precision meets a target at each DDIM step
- The resulting "DDfire" outperforms many state-of-the-art diffusion methods at total NFE budgets of 20, 100, 1000



# Noisy FFHQ results

- Results on $256 \times 256$ FFHQ faces with measurement noise $\sigma_w = 0.05$:

| | | Inpaint (box) | | | Deblur (Gaussian) | | | Deblur (Motion) | | | 4× Super-resolution | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # NFEs | Model | PSNR↑ | LPIPS↓ | FID↓ | PSNR↑ | LPIPS↓ | FID↓ | PSNR↑ | LPIPS↓ | FID↓ | PSNR↑ | LPIPS↓ | FID↓ |
| 20 | DiffPIR | 20.87 | 0.2741 | 41.50 | 23.55 | 0.3269 | 41.29 | 27.31 | 0.2704 | 29.27 | 22.32 | 0.3560 | 44.85 |
| | DDRM | **22.02** | 0.2052 | 40.61 | 26.27 | 0.2896 | 51.70 | - | - | - | **28.62** | **0.2417** | 45.82 |
| | DDfire | 21.80 | **0.1974** | **28.49** | **27.18** | **0.2843** | **36.22** | **28.52** | **0.2455** | **28.86** | 27.02 | 0.2917 | **37.72** |
| 100 | DiffPIR | 22.44 | 0.2415 | 31.98 | 24.57 | 0.2936 | 34.82 | 26.91 | 0.2683 | 26.67 | 26.76 | 0.3061 | 32.33 |
| | ΠGDM | 21.75 | 0.2614 | 44.41 | 24.34 | 0.3125 | 45.34 | 25.94 | 0.2706 | 41.95 | 25.42 | 0.3109 | 51.41 |
| | DDfire | **23.78** | **0.1623** | **26.75** | **27.48** | **0.2274** | **25.48** | **27.79** | **0.2193** | **25.91** | **27.20** | **0.2399** | **26.24** |
| 1000 | DPS | 22.84 | 0.1793 | 35.69 | 26.32 | 0.2327 | 25.18 | 27.64 | 0.2176 | 27.17 | 27.11 | 0.2360 | 27.38 |
| | DDfire | **24.14** | **0.1579** | **24.56** | **26.84** | **0.2259** | **24.68** | **27.71** | **0.2155** | **24.57** | **27.32** | **0.2356** | **25.75** |

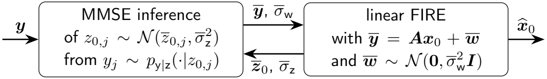
- DDfire outperforms the competitors in 33 of the 36 cases

# Extension to generalized-linear models

- To handle problems like phase retrieval, dequantization, Poisson regression, and non-Gaussian additive noise, we extend FIRE & DDfire to the generalized linear model (GLM):

$$\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{z}_0) = \prod_{j=1}^{m} p_{\mathsf{y}|\mathsf{z}}(y_j|z_{0,j}) \quad \text{with} \quad \boldsymbol{z}_0 \triangleq \boldsymbol{A}\boldsymbol{x}_0$$

- We do this using expectation-propagation iterations between linear FIRE and a scalar MMSE inference stage:



$\boldsymbol{y}$

MMSE inference of $z_{0,j} \sim \mathcal{N}(\overline{z}_{0,j}, \overline{\sigma}_{\mathsf{z}}^2)$ from $y_j \sim p_{\mathsf{y}|\mathsf{z}}(\cdot|z_{0,j})$

$\overline{\boldsymbol{y}}, \overline{\sigma}_{\mathsf{w}}$

$\overline{\boldsymbol{z}}_0, \overline{\sigma}_{\mathsf{z}}$

linear FIRE with $\overline{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{x}_0 + \overline{\boldsymbol{w}}$ and $\overline{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{0}, \overline{\sigma}_{\mathsf{w}}^2\boldsymbol{I})$

$\widehat{\boldsymbol{x}}_0$

- The GLM DDfire gives state-of-the-art noisy phase-retrieval performance with both oversampled Fourier (OSF) and coded diffraction pattern (CDP) operators:

| | | OSF | | | CDP | | |
|---|---|---|---|---|---|---|---|
| # NFEs | Model | PSNR↑ | LPIPS↓ | FID↓ | PSNR↑ | LPIPS↓ | FID↓ |
| - | HIO | 23.66 | 0.5299 | 130.58 | 17.59 | 0.5818 | 84.87 |
| 1000 | DOLPH | 14.73 | 0.7089 | 389.88 | 25.76 | 0.2163 | 32.93 |
| 1000 | DPS | 23.63 | 0.3326 | 53.91 | 29.19 | 0.1994 | 27.87 |
| 800 | prDeep | 30.90 | 0.1585 | 31.51 | 19.24 | 0.4352 | 59.44 |
| 800 | DDfire | **33.56** | **0.1160** | **28.94** | 30.01 | 0.1767 | 23.49 |
| 100 | DDfire | 25.88 | 0.2643 | 46.54 | **30.16** | **0.1707** | **23.30** |

# Outline

1. Probabilistic bounds on recovered-image accuracy
2. Probabilistic bounds for quantitative/task-based imaging
3. Efficient posterior sampling via diffusion with fast iterative renoising
4. Fast posterior sampling with regularized cGANs

# Fast posterior sampling for inverse problems

- Although diffusion-based posterior sampling has become popular, it's computationally challenging:
  - High accuracy methods consume hundreds of NFEs
- For supervised posterior sampling, which trains on a particular $\mathcal{A}(\cdot)$ operator, single-NFE sampling is possible:
  - Conditional VAEs, conditional GANs, conditional normalizing flows
- Conventional wisdom says cGANs have unstable training and poor diversity. But is this really true?
  - In our experience, training instability vanished with the Wasserstein GAN[1,2]
  - Poor diversity can indeed be a problem[3]
- Our work shows how appropriate regularization can solve the diversity problem and lead to single-NFE cGANs that are more accurate than modern diffusion-based samplers

[1] Arjovsky,Chintala,Bottou'17, [2] Gulrajani,Ahmed,Arjovsky,Dumoulin,Courville'17, [3] Adler,Öktem'18

# Our approach

The regularized Wasserstein cGAN:

- Generator $G_{\boldsymbol{\theta}}$: outputs $\widehat{\boldsymbol{x}}_i = G_{\boldsymbol{\theta}}(\boldsymbol{z}_i, \boldsymbol{y})$ for code realization $\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
- Discriminator $D_{\boldsymbol{\phi}}$: aims to distinguish true $(\boldsymbol{x}, \boldsymbol{y})$ from fake $(\widehat{\boldsymbol{x}}_i, \boldsymbol{y})$
- Training:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \Big\{ \mathbb{E}_{\mathsf{x,z,y}} \{ D_{\boldsymbol{\phi}}(\boldsymbol{x}_0, \boldsymbol{y}) - D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{y}), \boldsymbol{y}) \} + \mathcal{R}(\boldsymbol{\theta}) - \mathcal{L}_{\mathsf{gp}}(\boldsymbol{\phi}) \Big\}$$

- The regularization $\mathcal{R}(\boldsymbol{\theta})$ is designed to enhance the fidelity & diversity of the generated samples

Contributions:

- rcGAN[1] designs $\mathcal{R}(\cdot)$ to reward correctness in the conditional mean and conditional trace-covariance
- pcaGAN[2] adds correctness in the $K$ principal eigencomponents of the conditional covariance matrix

[1] Bendel,Ahmad,S'22, [2] Bendel,Ahmad,S'23

# rcGAN

rcGAN[1] regularizes using an L1 penalty and a standard-deviation reward:

$$\mathcal{R}_{\text{rc}}(\boldsymbol{\theta}) \triangleq \underbrace{\mathbb{E}_{\mathsf{x},\mathsf{z}_1,\dots,\mathsf{z}_P,\mathsf{y}}\left\{\|\boldsymbol{x}_0 - \widehat{\boldsymbol{x}}_{(P)}\|_1\right\}}_{\triangleq\, \mathcal{L}_{1,P}(\boldsymbol{\theta})} - \beta_{\text{std}} \underbrace{\textstyle\sum_{i=1}^{P} \mathbb{E}_{\mathsf{z}_1,\dots,\mathsf{z}_P,\mathsf{y}}\left\{\|\widehat{\boldsymbol{x}}_i - \widehat{\boldsymbol{x}}_{(P)}\|_1\right\}}_{\triangleq\, \mathcal{L}_{\text{std},P}(\boldsymbol{\theta})}$$

where $\widehat{\boldsymbol{x}}_{(P)} \triangleq \frac{1}{P}\sum_{i=1}^{P} \widehat{\boldsymbol{x}}_i$ is the average of $P$ posterior samples.

Key points:

- $\beta_{\text{std}}$ controls the amount of diversity, and is automatically optimized during training
- Can prove[1] $\mathcal{R}_{\text{rc}}(\boldsymbol{\theta})$ yields $\mathbb{E}\{\widehat{\boldsymbol{x}}_i|\boldsymbol{y}\} = \mathbb{E}\{\boldsymbol{x}_0|\boldsymbol{y}\}$ and $\operatorname{tr}\operatorname{Cov}\{\widehat{\boldsymbol{x}}_i|\boldsymbol{y}\} = \operatorname{tr}\operatorname{Cov}\{\boldsymbol{x}_0|\boldsymbol{y}\}$ with Gaussian $\boldsymbol{x}_0|\boldsymbol{y}$
- Can prove[1] $\mathcal{R}(\boldsymbol{\theta}) = \mathcal{L}_{2,P}(\boldsymbol{\theta}) - \beta_{\text{var}}\mathcal{L}_{\text{var},P}(\boldsymbol{\theta})$ does not (for any $\beta_{\text{var}}$)

[1] Bendel,Ahmad,S'22

# pcaGAN

Goal: Ensure that $\widehat{\boldsymbol{v}}_k = \boldsymbol{v}_k$ and $\widehat{\lambda}_k = \lambda_k$ for $k = 1, \dots, K$

- $\{(\widehat{\boldsymbol{v}}_k, \widehat{\lambda}_k)\}_{k=1}^K$ are the principal evecs/evals of $\mathrm{Cov}\{\widehat{\boldsymbol{x}}_i | \boldsymbol{y}\}$
- $\{(\boldsymbol{v}_k, \lambda_k)\}_{k=1}^K$ are the principal evecs/evals of $\mathrm{Cov}\{\boldsymbol{x}_0 | \boldsymbol{y}\}$
- $K$ is user-specified

We propose

$$\mathcal{R}_{\text{pca}}(\boldsymbol{\theta}) \triangleq \mathcal{R}_{\text{rc}}(\boldsymbol{\theta}) + \beta_{\text{pca}} \mathcal{L}_{\text{evec}}(\boldsymbol{\theta}) + \beta_{\text{pca}} \mathcal{L}_{\text{eval}}(\boldsymbol{\theta}),$$

where

$$\mathcal{L}_{\text{evec}}(\boldsymbol{\theta}) \triangleq -\mathbb{E}_{\mathsf{y}} \left\{ \mathbb{E}_{\mathsf{x},\mathsf{z}_1,\dots,\mathsf{z}_P|\mathsf{y}} \left\{ \textstyle\sum_{k=1}^K [\widehat{\boldsymbol{v}}_k(\boldsymbol{\theta})^\top (\boldsymbol{x}_0 - \boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}})]^2 \middle| \boldsymbol{y} \right\} \right\}$$

$$\mathcal{L}_{\text{eval}}(\boldsymbol{\theta}) \triangleq \mathbb{E}_{\mathsf{y}} \left\{ \mathbb{E}_{\mathsf{x},\mathsf{z}_1,\dots,\mathsf{z}_P|\mathsf{y}} \left\{ \textstyle\sum_{k=1}^K \left(1 - \lambda_k / \widehat{\lambda}_k(\boldsymbol{\theta})\right)^2 \middle| \boldsymbol{y} \right\} \right\}$$

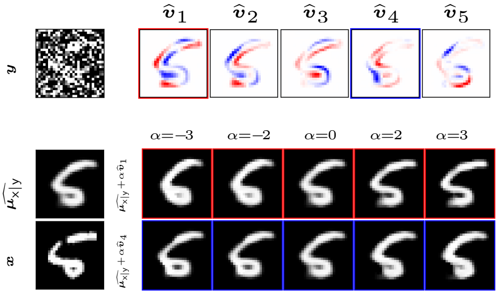
and where

the unknown $\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}} \triangleq \mathbb{E}\{\boldsymbol{x}_0 | \boldsymbol{y}\}$ and $\{\lambda_k\}_{k=1}^K$ are approximated using stopgrad and an SVD

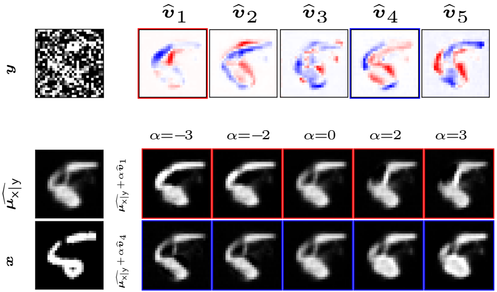
# MNIST denoising: Visualizing the principal uncertainty components

- Principal eigenvectors $\{\boldsymbol{v}_k\}_{k=1}^{K}$ are shown below for $K = 5$
- Also shown are $\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}} \pm \alpha\widehat{\boldsymbol{v}}_k$ for $k \in \{1,4\}$ and $\alpha \in \{-3,-2,0,2,3\}$

**pcaGAN:**


$\widehat{\boldsymbol{v}}_1$
$\widehat{\boldsymbol{v}}_2$
$\widehat{\boldsymbol{v}}_3$
$\widehat{\boldsymbol{v}}_4$
$\widehat{\boldsymbol{v}}_5$
$\boldsymbol{y}$
$\alpha{=}{-3}$
$\alpha{=}{-2}$
$\alpha{=}0$
$\alpha{=}2$
$\alpha{=}3$
$\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}}$
$\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}} + \alpha\widehat{\boldsymbol{v}}_1$
$\boldsymbol{x}$
$\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}} + \alpha\widehat{\boldsymbol{v}}_4$

**NPPC:**[1]


$\widehat{\boldsymbol{v}}_1$
$\widehat{\boldsymbol{v}}_2$
$\widehat{\boldsymbol{v}}_3$
$\widehat{\boldsymbol{v}}_4$
$\widehat{\boldsymbol{v}}_5$
$\boldsymbol{y}$
$\alpha{=}{-3}$
$\alpha{=}{-2}$
$\alpha{=}0$
$\alpha{=}2$
$\alpha{=}3$
$\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}}$
$\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}} + \alpha\widehat{\boldsymbol{v}}_1$
$\boldsymbol{x}$
$\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}} + \alpha\widehat{\boldsymbol{v}}_4$

- pcaGAN's eigenvectors show much more meaningful structure

[1] Nehme,Yair,Michaeli'23

# Large-scale image completion/inpainting

We inpainted large random masks on 256x256 FFHQ face images

- Results (20k test images):

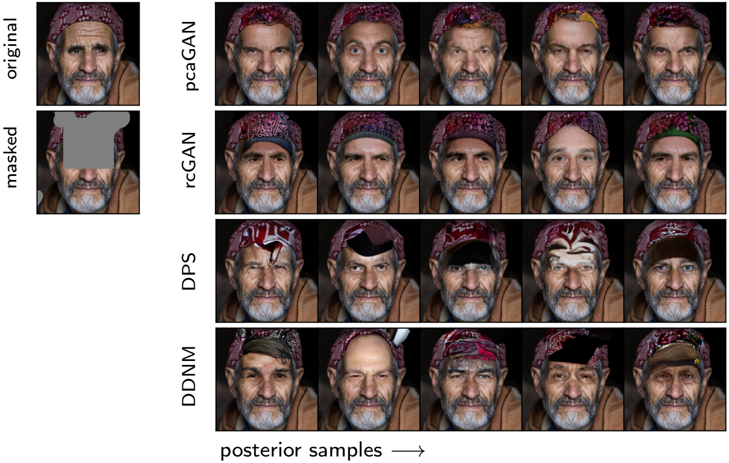
| Model | CFID↓ | FID↓ | LPIPS↓ | Time (40 samples)↓ |
|---|---|---|---|---|
| DPS[1] (1000 NFEs) | 7.26 | 2.00 | 0.1245 | 14 min |
| DDNM[2] (100 NFEs) | 11.30 | 3.63 | 0.1409 | 30 s |
| DDRM[3] (20 NFEs) | 13.17 | 5.36 | 0.1587 | 5 s |
| pscGAN[4] | 18.44 | 8.40 | 0.1716 | **325 ms** |
| CoModGAN[5] | 7.85 | 2.23 | 0.1290 | **325 ms** |
| rcGAN | 7.51 | 2.12 | 0.1262 | **325 ms** |
| pcaGAN ($K = 2$) | **7.08** | **1.98** | **0.1230** | **325 ms** |

- pcaGAN outperformed all diffusion *and* cGAN competitors
- cGANs are 15x to 2500x faster than the diffusion methods

[1] Chung et al'23, [2] Wang et al'23, [3] Kawar et al'22, [4] Ohayon et al'21, [5] Zhao et al'21

# Example FFHQ inpainting

rcGAN and pcaGAN generate samples that are both high quality and diverse



# Results on accelerated MR image recovery

We reconstructed multicoil fastMRI[1] T2 brain images at acceleration $R = 8$

- Results (74 test images):

| Model | CFID↓ | FID↓ | PSNR↑ | SSIM↑ | LPIPS↓ | DISTS↓ | Time (4 samples)↓ |
|---|---|---|---|---|---|---|---|
| E2E-VarNet[2] | 36.86 | 44.04 | **36.49** | 0.9220 | 0.0575 | 0.1253 | 316ms |
| Langevin (Jalal[3]) | 48.59 | 52.62 | 33.90 | 0.9137 | 0.0579 | 0.1086 | 14 min |
| cGAN (Adler[4]) | 59.94 | 31.81 | 33.51 | 0.9111 | 0.0614 | 0.1252 | **217 ms** |
| pscGAN[5] | 39.67 | 43.39 | 34.92 | 0.9222 | 0.0532 | 0.1128 | **217 ms** |
| rcGAN | <u>24.04</u> | <u>28.43</u> | 35.42 | <u>0.9257</u> | <u>0.0379</u> | <u>0.0877</u> | **217 ms** |
| pcaGAN ($K = 1$) | **21.65** | **28.35** | <u>35.94</u> | **0.9283** | **0.0344** | **0.0799** | **217 ms** |

- pcaGAN won in all metrics but PSNR
- The cGANs generated samples 3800x faster than the Langevin method
- Note: PSNR, SSIM, LPIPS, DISTS computed using $\widehat{\boldsymbol{x}}_{(P)}$ for the optimal $P$

[1] Zbontar et al'18, [2] Sriram et al'19, [3] Jalal et al'21, [4] Adler,Öktem'18, [5] Ohayon et al'21

# Conclusion

- Due to ill-posedness and the possibility of hallucinations, there's a need for inverse-problem solvers with performance guarantees

- By combining approximate posterior sampling with conformal prediction, we proposed image-recovery and quantitative-imaging methods with probabilistic guarantees

- For the unsupervised scenario, we presented a new diffusion posterior sampler, based on iterative renoising, with SOTA performance over a wide range of NFEs

- For the supervised case, we presented a regularized cGAN sampler that is more accurate than contemporary diffusion methods while consuming only a single NFE