pcaGAN: Improving posterior-sampling cGANs via principal component regularization

Matthew Bendel[*], Rizwan Ahmad[‡], and Philip Schniter[*]

[*]Dept. ECE, The Ohio State University, {bendel.8,schniter.1}@osu.edu

[‡]Dept. BME, The Ohio State University, ahmad.46@osu.edu

THE OHIO STATE UNIVERSITY

Supported in part by the NIH under grant R01-EB029957

Asilomar Conference on Signals, Systems, and Computers
October 2024

Motivation: Imaging inverse problems

Goal: Recover image $\boldsymbol{x}$ from measurements $\boldsymbol{y} = \mathcal{M}(\boldsymbol{x})$:

- $\mathcal{M}(\cdot)$ masks, distorts, and/or corrupts $\boldsymbol{x}$ with noise.
- Applications: inpainting, super-resolution, deblurring, computed tomography, magnetic resonance imaging (MRI), etc.
- Solution typically posed as point estimation: find the single "best" $\widehat{\boldsymbol{x}}$

Challenges with point-estimation:

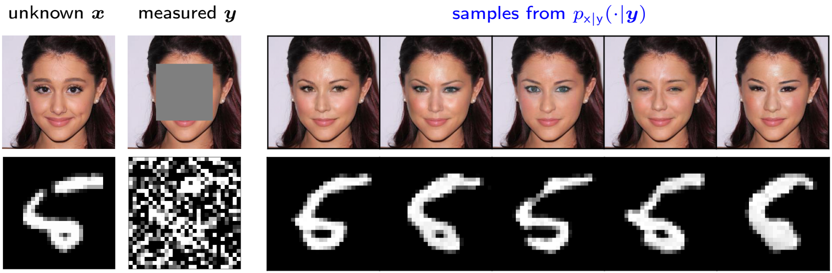
- Doesn't provide uncertainty quantification
- Doesn't navigate the perception-distortion tradeoff[1]

Solution: Sample from posterior distribution $p_{\mathsf{x}|\mathsf{y}}(\boldsymbol{x}|\boldsymbol{y}) = \dfrac{p_{\mathsf{y}|\mathsf{x}}(\boldsymbol{y}|\boldsymbol{x})p_{\mathsf{x}}(\boldsymbol{x})}{\int p_{\mathsf{y}|\mathsf{x}}(\boldsymbol{y}|\boldsymbol{x})p_{\mathsf{x}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}$

[1] Blau,Michaeli'18

Posterior sampling

The posterior distribution $p_{\mathsf{x}|\mathsf{y}}(\cdot|\boldsymbol{y})$ represents all knowledge about the image $\boldsymbol{x}$ given the corrupted measurements $\boldsymbol{y}$



Typical posterior-sampling methods in imaging:

- cVAEs, cNFs, cGANs: Fast but inaccurate?
- Diffusion methods: Slow but accurate?

Our approach

We build on rcGAN[1], a type of Wasserstein cGAN:

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

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

- rcGAN's regularization $\mathcal{R}(\boldsymbol{\theta})$ rewards correctness in conditional mean and conditional trace-covariance

Contributions:

- A new $\mathcal{R}(\boldsymbol{\theta})$ that **also** enforces correctness in the K principal components of the conditional covariance matrix
- We call our approach pcaGAN

[1] Bendel,Ahmad,Schniter'22

Background on rcGAN

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

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

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:

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

[1] Bendel,Ahmad,Schniter'22

The proposed pcaGAN

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

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

We propose

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

where

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

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

and

we approximate the unknown $\{(\boldsymbol{v}_k, \lambda_k)\}_{k=1}^K$ and $\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}} \triangleq \mathbb{E}\{\boldsymbol{x} | \boldsymbol{y}\}$

Understanding pcaGAN

Eigenvector regularization:

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

- If $\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}} = \mathbb{E}\{\boldsymbol{x}|\boldsymbol{y}\}$ was known, minimizing over $\boldsymbol{\theta}$ would force $\{\widehat{\boldsymbol{v}}_k = \boldsymbol{v}_k\}_{k=1}^K$
- We set $\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}} \approx \widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}} = \texttt{StopGrad}(\widehat{\boldsymbol{x}}_{(P_{\text{pca}})})$ after $\widehat{\boldsymbol{x}}_{(P_{\text{pca}})}$ stabilizes
- We compute $\{\widehat{\boldsymbol{v}}_k\}_{k=1}^K$ using an SVD of $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^{P_{\text{pca}}}$, where $P_{\text{pca}} \approx 10K$

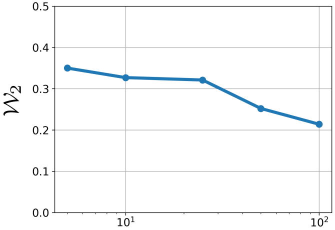
Eigenvalue regularization:

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

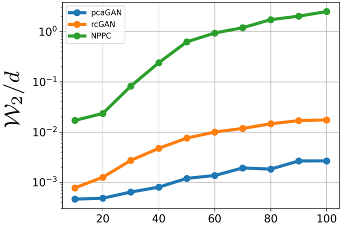
- If λ_k was known, minimizing over $\boldsymbol{\theta}$ would force $\{\widehat{\lambda}_k = \lambda_k\}_{k=1}^K$
- We set $\lambda_k \approx \texttt{StopGrad}(\frac{1}{P_{\text{pca}}+1}\|\widehat{\boldsymbol{v}}_k^{\mathsf{T}}[\boldsymbol{x}-\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}}, \widehat{\boldsymbol{x}}_1-\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}}, \dots, \widehat{\boldsymbol{x}}_{P_{\text{pca}}}-\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}}]\|_2^2)$ after $\{\widehat{\boldsymbol{v}}_k\}$ stabilize, which is correct in expectation when $\widehat{\boldsymbol{v}}_k$ and $\widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}}$ are correct
- We compute $\{\widehat{\lambda}_k\}_{k=1}^K$ using the same SVD as above

Gaussian toy experiments

- Here we recover $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathsf{x}}, \boldsymbol{\Sigma}_{\mathsf{x}})$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$ with inpainting $\boldsymbol{A} \in \mathbb{R}^{\frac{d}{2} \times d}$ and $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. Both $\boldsymbol{\mu}_{\mathsf{x}}$ and $\boldsymbol{\Sigma}_{\mathsf{x}}$ are random
- Performance measured via Wasserstein-2 distance $\mathcal{W}_2(p_{\mathsf{x}|\mathsf{y}}, p_{\widehat{\mathsf{x}}|\mathsf{y}})$



$\#$ components K when $d = 100$



signal dimension d when $K = d$

- $\mathcal{W}_2$ decreases with K, and pcaGAN beats both rcGAN and NPPC[1] in $\mathcal{W}_2$
  - NPPC trains a neural net to directly estimate $\{(\lambda_k, \boldsymbol{v}_k)\}_{k=1}^K$ from $\boldsymbol{y}$

[1] Nehme,Yair,Michaeli'23

MNIST denoising experiments

We recovered MNIST digits $\boldsymbol{x}$ from measurements $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{w}$ with $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

- We measured by rMSE, residual error magnitude (REM) at $K = 5$:

$$\text{REM}_5 \triangleq \mathbb{E}_{\mathsf{x},\mathsf{y}} \left\{ \|(\boldsymbol{I} - \widehat{\boldsymbol{V}}_5 \widehat{\boldsymbol{V}}_5^{\mathsf{T}})(\boldsymbol{x} - \widehat{\boldsymbol{\mu}_{\mathsf{x}|\mathsf{y}}})\|_2 \right\} \text{ where } \widehat{\boldsymbol{V}}_5 \triangleq [\widehat{\boldsymbol{v}}_1, \dots, \widehat{\boldsymbol{v}}_5]$$

- We also measured Conditional FID (CFID)[1], which is similar to FID but measures the discrepancy between $p_{\widehat{\mathsf{x}}|\mathsf{y}}$ and $p_{\mathsf{x}|\mathsf{y}}$ (not between $p_{\widehat{\mathsf{x}}}$ and p_{x})

- Results (128 test images):

Model	rMSE↓	REM$_5$↓	CFID↓	Time (128 samples)↓
NPPC ($K = 5$)	**3.94**	3.63	–	**112 ms**
rcGAN	4.04	<u>3.41</u>	<u>63.44</u>	<u>118</u> ms
pcaGAN (ours, $K = 5$)	<u>4.02</u>	**3.31**	**61.48**	<u>118</u> ms

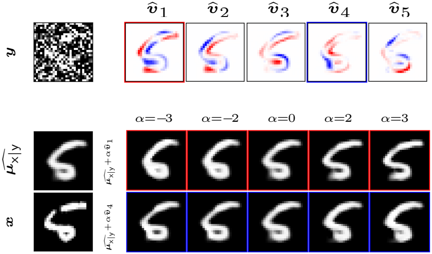
- pcaGAN wins in both REM and CFID (note NPPC is not generative)

[1] Soloveitchik'21

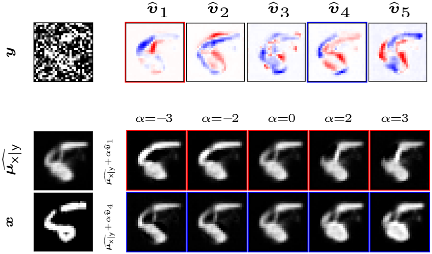
MNIST: visualizing the principal uncertainty components

- 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:



NPPC:



- pcaGAN's eigenvectors show much more meaningful structure

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)	<u>7.26</u>	<u>2.00</u>	<u>0.1245</u>	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 (ours, $K = 2$)	**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

pcaGAN generates samples that are both high quality and diverse

Results on accelerated MR image recovery

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

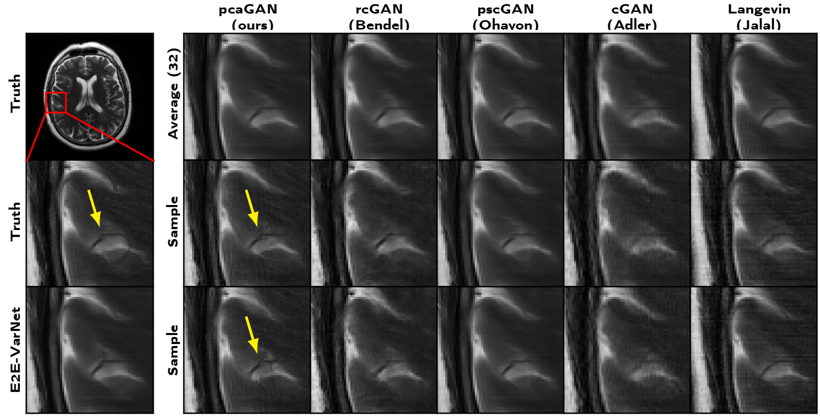
- Results (74 test images):

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	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 (ours, $K = 1$)	**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

Example MRI recoveries



- posterior samples from pcaGAN show meaningful variations (see arrows)
- posterior samples from pscGAN show no variation
- posterior samples from cGAN (Adler) and Langevin (Jalal) show unwanted artifacts

Summary

Goal:

- Fast and accurate posterior sampling for imaging inverse problems

Contribution:

- A cGAN with a regularization that encourages correctness in the $\boldsymbol{y}$-conditional mean, trace-covariance, and K principal covariance components

Experimental results

- Considered MNIST denoising, FFHQ inpainting, and multicoil MRI
- Outperforms existing cGANs and diffusion methods (DPS, DDRM, DDNM, Langevin) in PSNR, SSIM, LPIPS, DISTS, FID, and CFID
- Runs 10x to 4000x faster than diffusion methods
- Outperforms NNPC (direct estimation of principal covariance components)