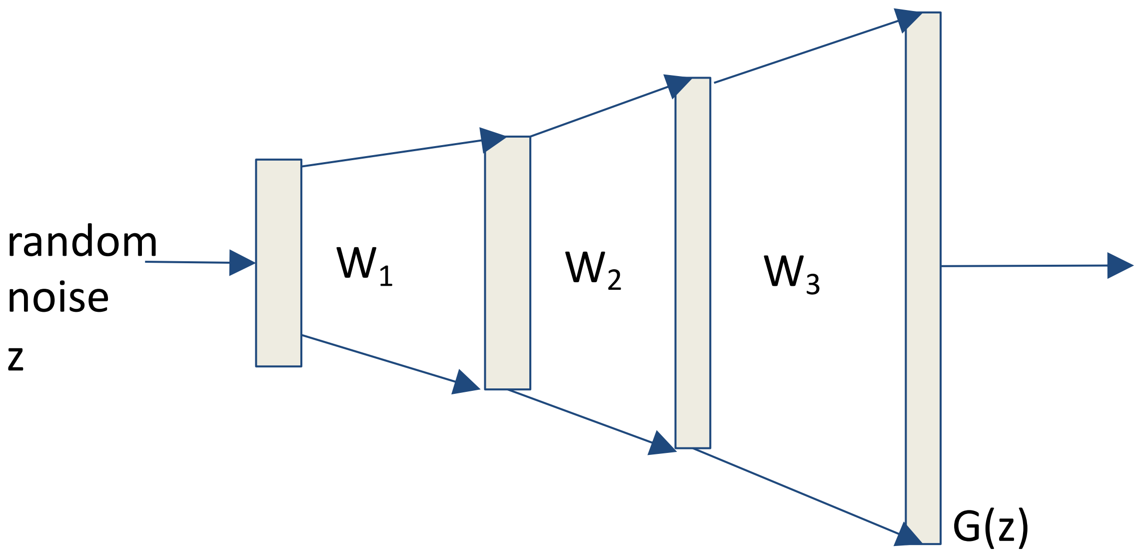
Deep Generative models And Inverse problems

Alex Dimakis, UT Austin

joint work with
Ajil Jalal, Sushrut Karmalkar, Joseph Dean, Giannis Daras, Qi Lei, Ashish Bora, Marius Arvinte, Jon Tamir and Eric Price, UT Austin

@AlexGDimakis

Deep generative models

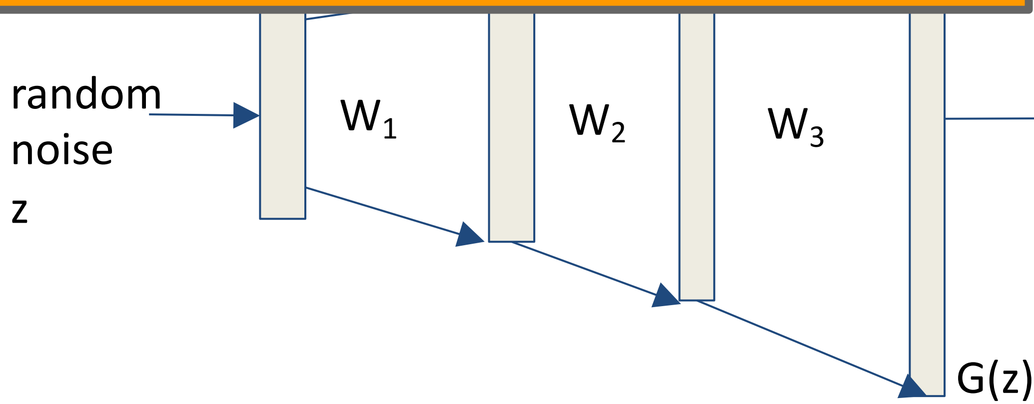


A DGM is a function $G(z) : \mathbb{R}^k \longrightarrow \mathbb{R}^n$

1. Differentiable a.e. (piecewise linear for ReLU activations)
2. Learned from samples

N looks like

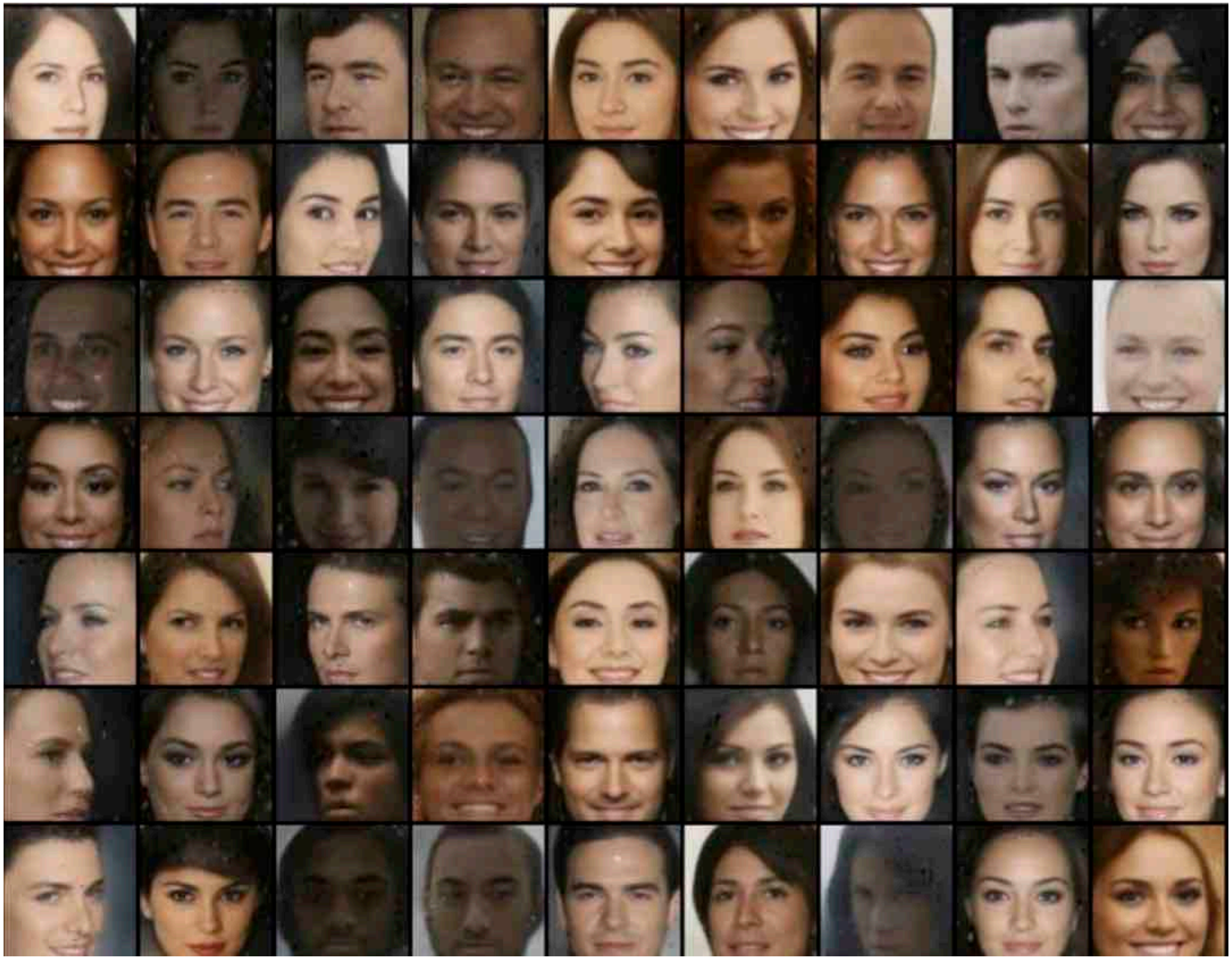
Any Resemblance
to Actual Persons,
Living or Dead,
is Purely Coincidental



A DGM is a function $G(z) : \mathbb{R}^k \longrightarrow \mathbb{R}^n$

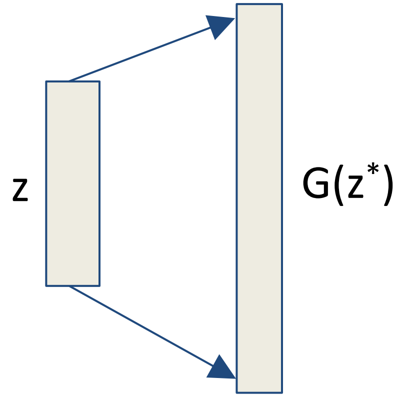
1. Differentiable a.e. (piecewise linear for ReLU activations)
2. Learned from samples

Empirically, Deep Generative models produce amazing images



Ok, Modern deep generative models produce amazing pictures.

But what can we do with them ?



Ok, Modern deep generative models produce amazing pictures.

But what can we do with them ?

A: generate fake pics for fake social media accounts lol

Ok, Modern deep generative models produce amazing pictures.

But what can we do with them ?

~~A: generate fake pics for fake social media accounts lol~~

They are **modular differentiable priors** that learn the statistics of your dataset.

Ok, Modern deep generative models produce amazing pictures.

But what can we do with them ?

A: We can solve **inverse problems:**

Denoising, Compression, Inpainting, Colorization, Compressed Sensing, Source separation, MRI, Phase Retrieval, Seismic Imaging, Anomaly detection, etc…

Talk Outline

Deep Generative models for inverse problems:

Compressed sensing using Generative models (CSGM)

(Bora et al. ICML 2017)

1. **Unsupervised** way to solve inverse problems using a deep generative model G(z)
2. CSGM Guarantees through Set Restricted Eigenvalue conditions.

Theory for Inverting deep generative models.

1.P. Hand and V. Voroninski, Global guarantees for enforcing deep priors by empirical risk, (COLT 2018)
2.Inverting Deep Generative models, One layer at a time
Q. Lei, A. Jalal, I. Dhillon, A.G.D, NeurIPS 2019
Daskalakis, Dhruv, and Zampetakis. "Constant-expansion suffices for compressed sensing with generative priors", 2021

A New Algorithmic idea: ILO: Intermediate Layer Optimization

Intermediate Layer Optimization for Inverse Problems using Deep Generative Models
ICML 2021, Deep-Inverse Workshop, NeurIPS 2020. Daras, Dean, Jalal, AD.
https://github.com/giannisdaras/ilo
https://arxiv.org/abs/2102.07364

New results: Robust Compressed Sensing MRI with Deep Generative Priors (NeurIPS'21)

Also if we have time (we won't): How to turn people into frogs ,
Bias and fairness in inverse problems

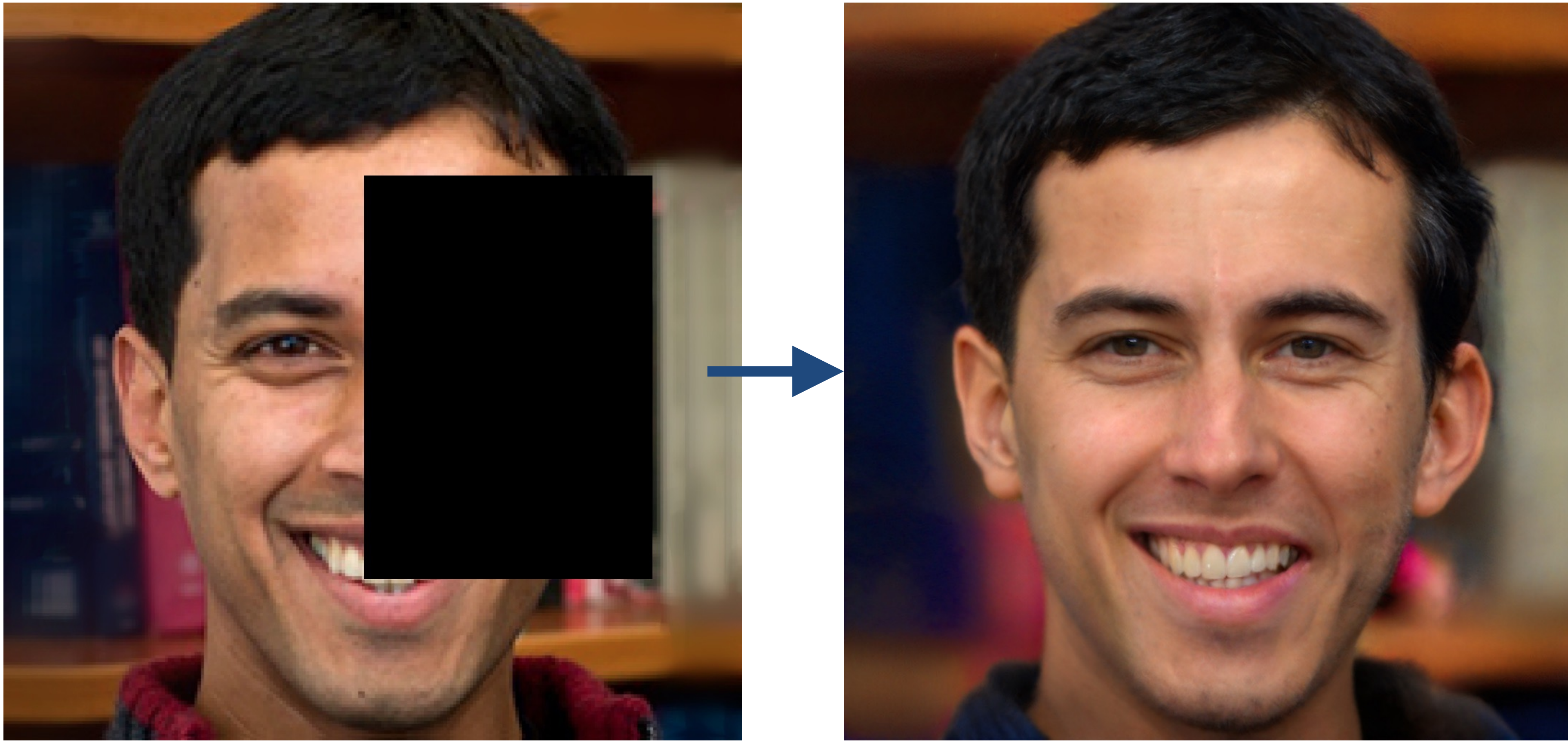
Fresh results- Inpainting

225x225

Inpainted Image

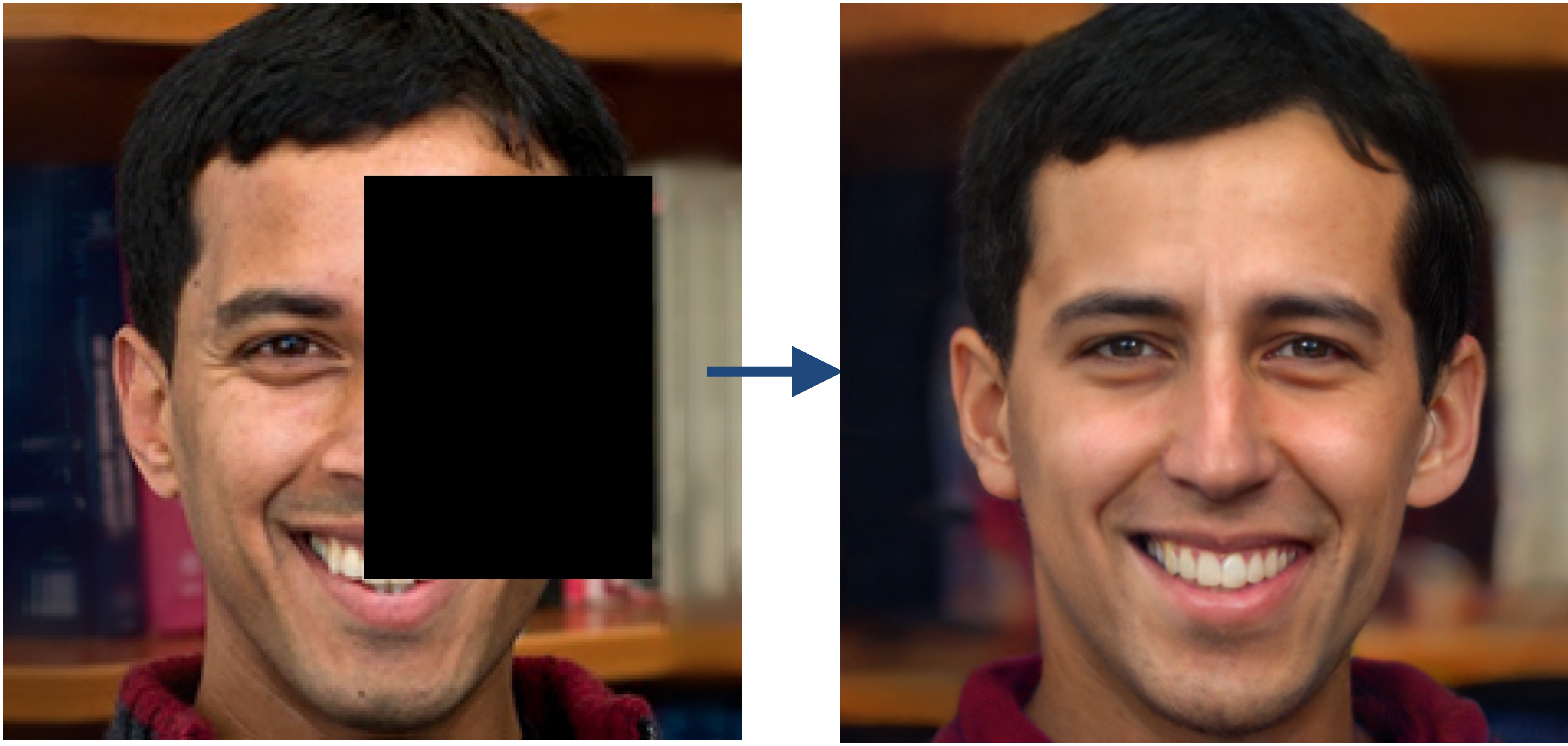
1024x1024

Fresh results- Inpainting



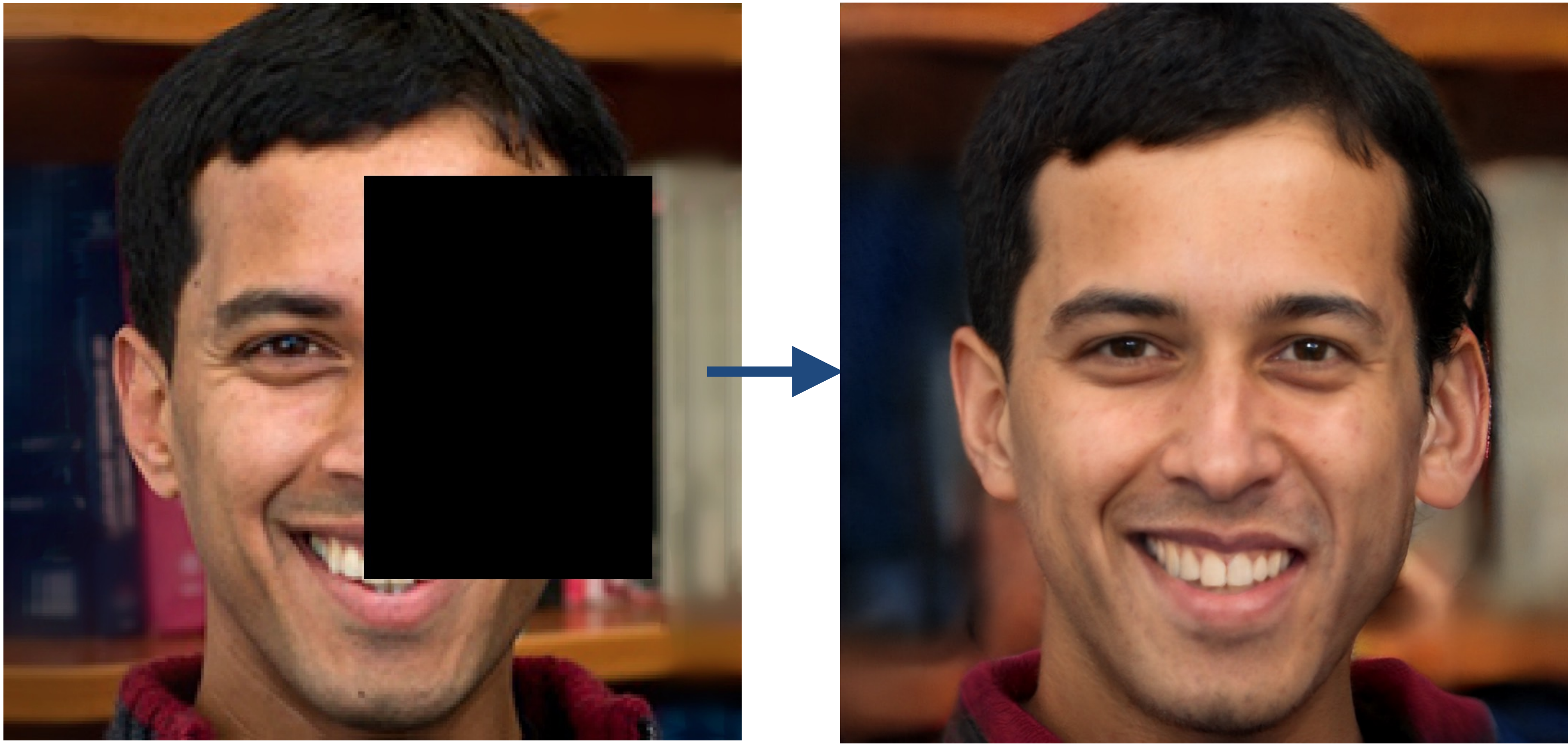
Compressed Sensing using Generative Models (CSGM)
by Bora et al. 2017 / PULSE using StyleGAN2 as a prior.

Fresh results- Inpainting



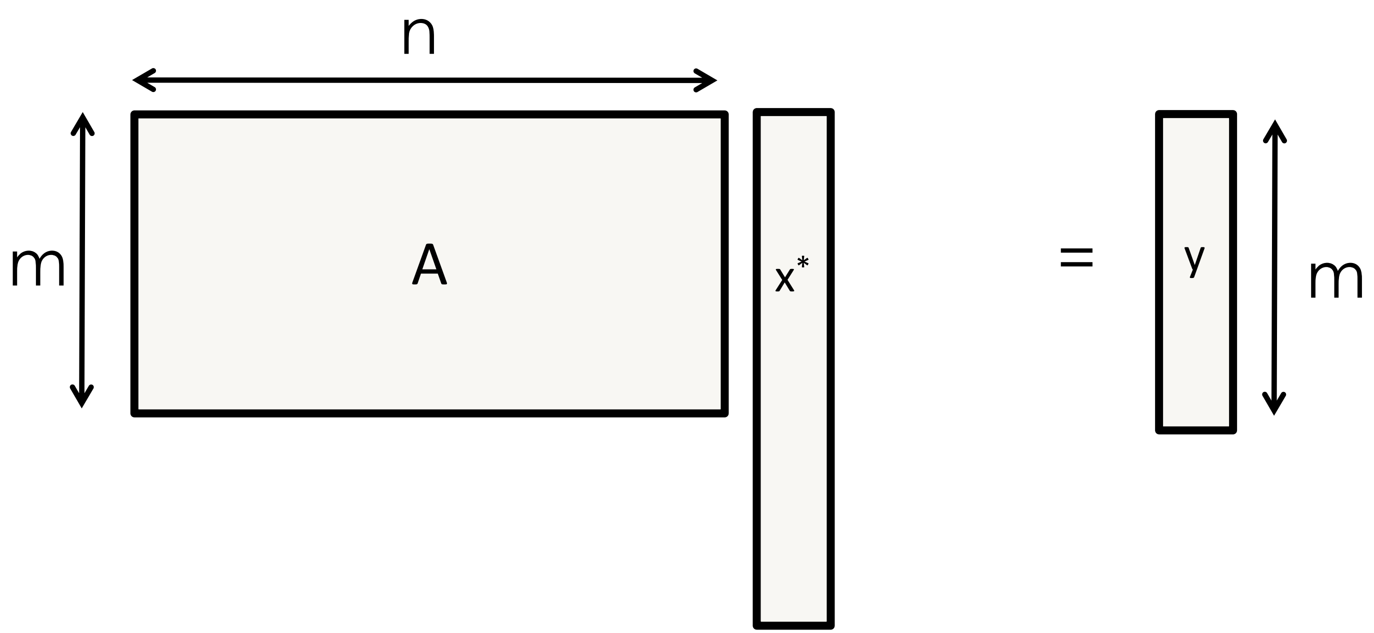
Intermediate Layer Optimization (ILO), Layers 1-2,
Prior distribution used: StyleGAN2

Fresh results- Inpainting

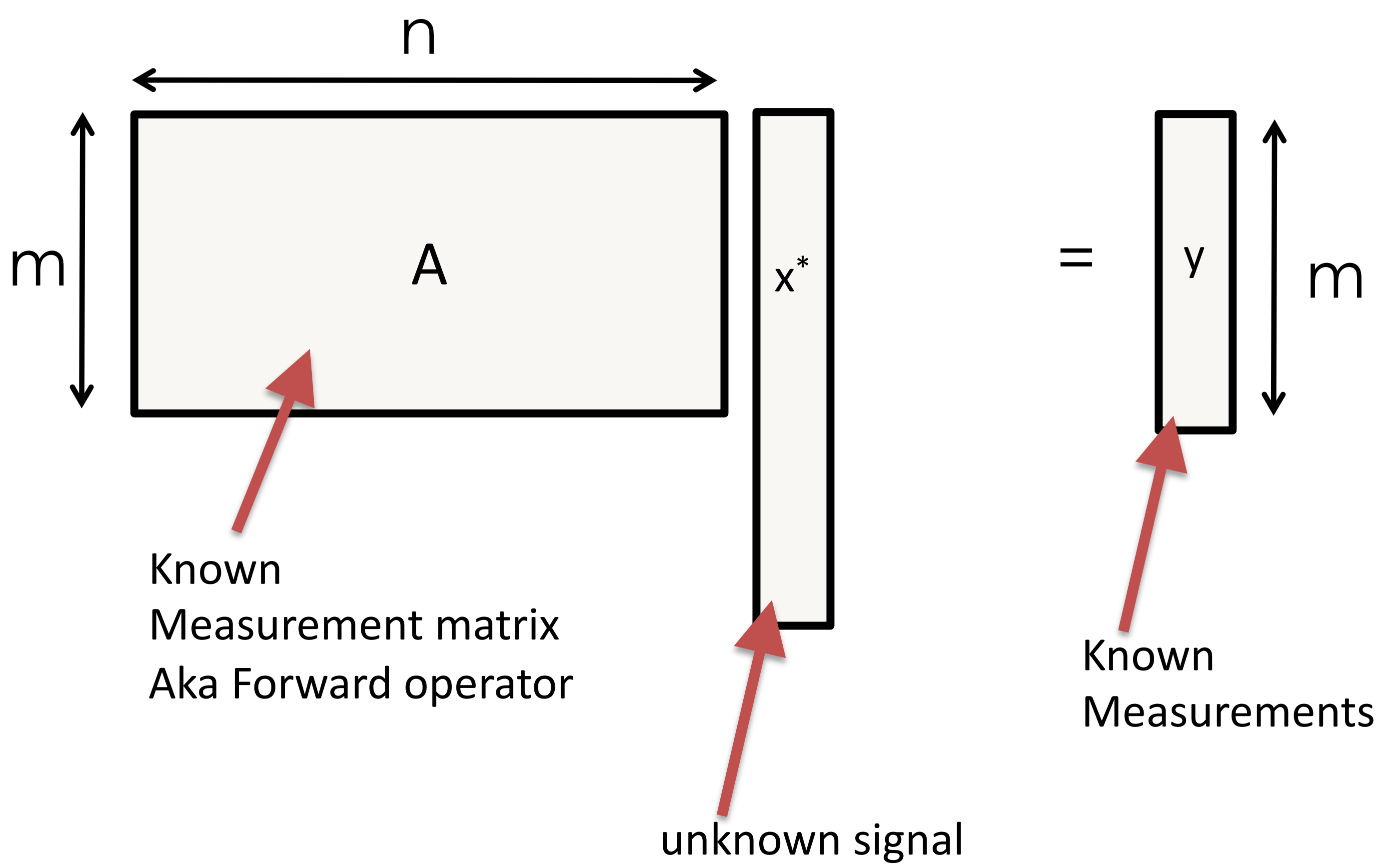


Intermediate Layer Optimization (ILO), Layers 1-4,
Prior distribution used: StyleGAN2

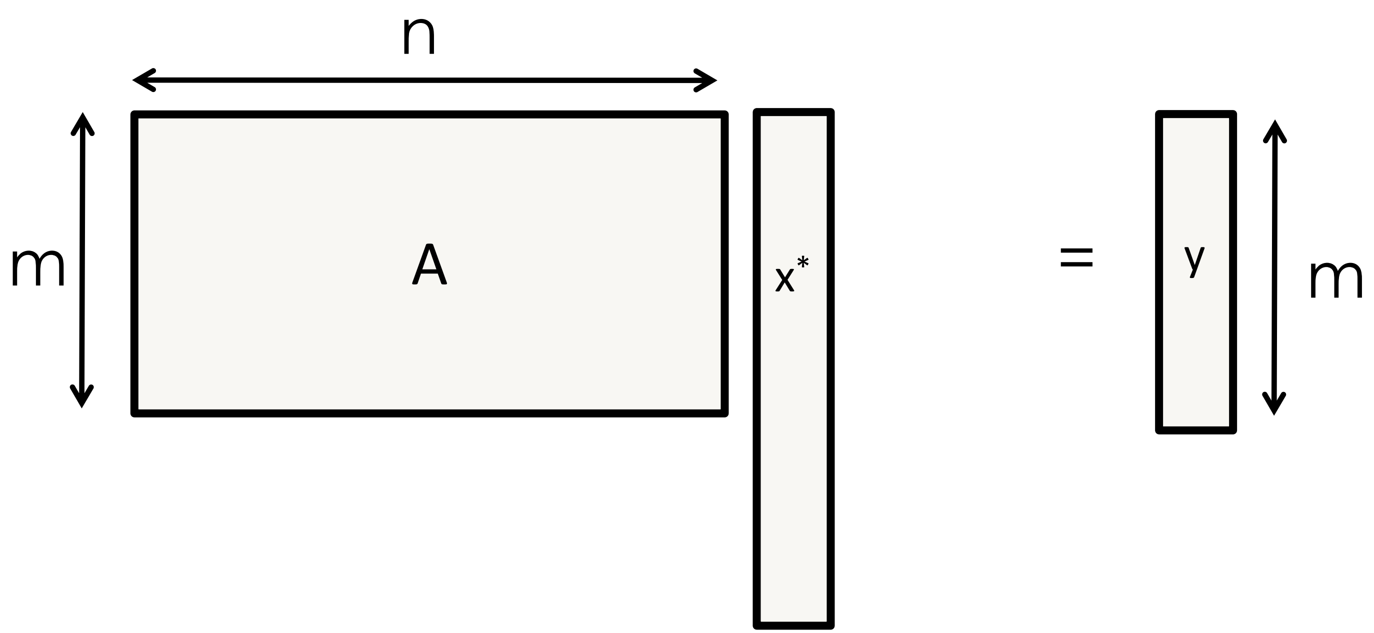
Compressed sensing= Linear Inverse problem



Compressed sensing= Linear Inverse problems



Compressed sensing= Linear Inverse problems



- You observe $y = A x^*$, *x in R^n , y in R^m, n>m*
- i.e. m (noisy) linear observations of an unknown vector y in R^n
- Goal: Recover x^* from y
- ill-posed: there are many possible x* that explain the measurements since we have m linear equations with n unknowns.
- High-dimensional statistics: Number of parameters n > number of samples m
- **Must make some assumption**: that x* **has some structure**
- ~~**x* is sparse**~~ → **x* is near the range of a pre-trained generator**

General setup: Linear Inverse problems

- $y = Ax^* + \text{noise}$
- $\min_x ||Ax - y|| + R(x)$

- Sparsity prior: $R(x) = ||x||_1$ (Lasso) or $||Dx||1$ (Lasso in DCT/Wavelet)

- $\min_z ||A\,G(z) - y||$ (CSGM)

- $R(x) = +\infty$ if x not in range of G(z)
  Otherwise Uniform over all x in range of generator

~~Sparsity~~ in compressed sensing

Sparsity in a basis
is a *decent* model for natural images

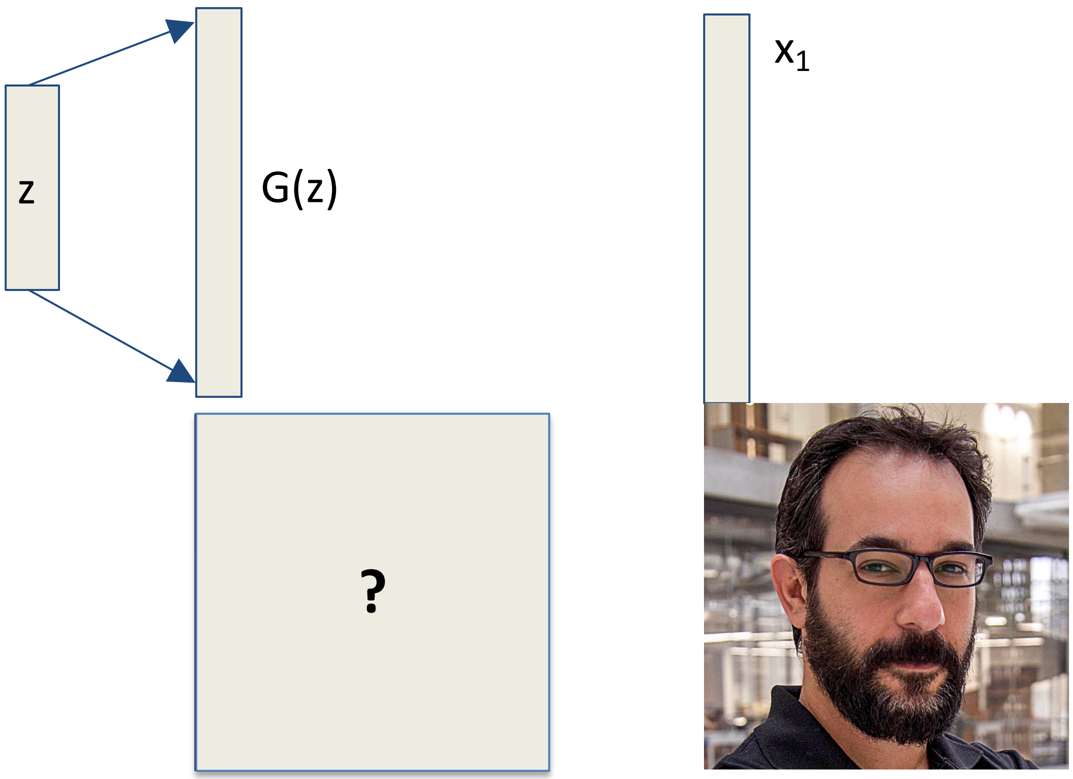
But now **we have much better data driven models for structure in high-dimensional distributions: DGMs**

Idea: Replace:
“ x is k-sparse “
"x is in the range of a deep generative model G(z)”

(Recent fact: this is a proper generalization: you can create all k-sparse vectors with a 2-layer network).
(Akshay Kamath, Sushrut Karmalkar, Eric Price,
Lower Bounds for Compressed Sensing with Generative Models, Arxiv)

How do we solve inverse problems?

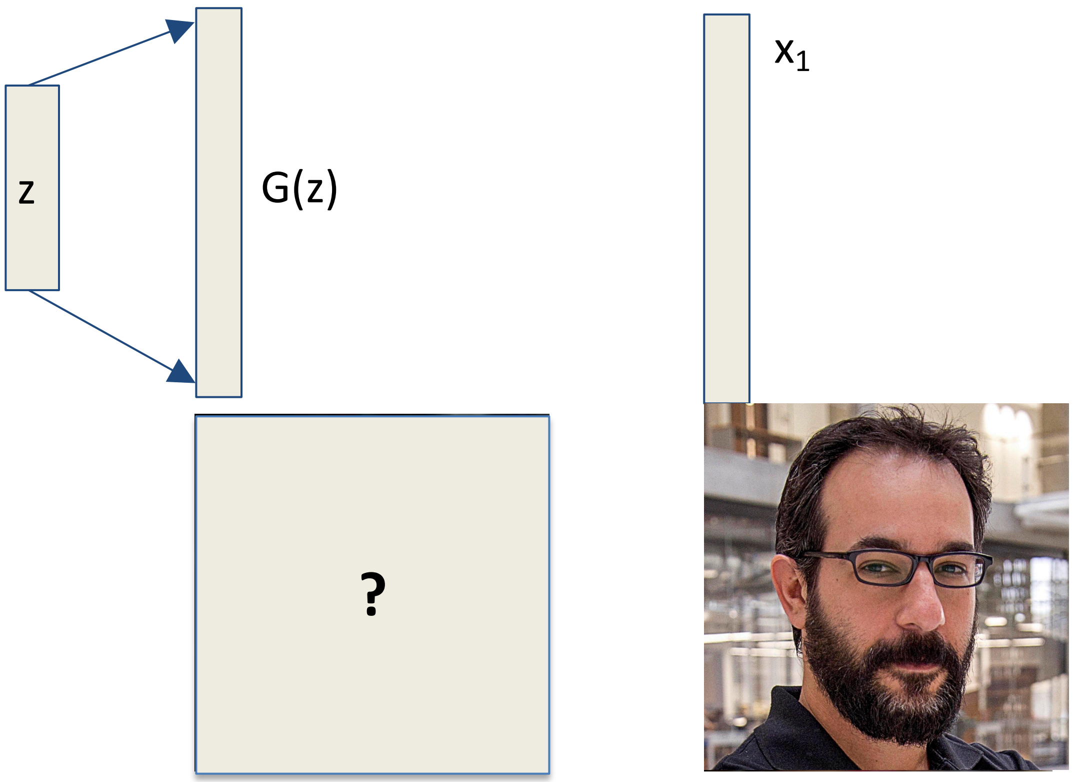
Simplest Inverse problem: Inverting a Generator



- Given a target image x_1 how do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 ?

Compressed sensing using generative models, Bora et al. ICML 2017.

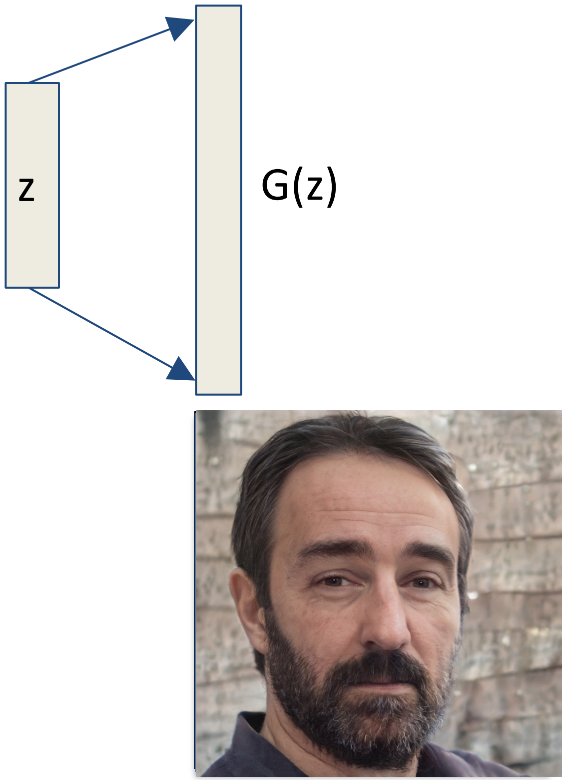
Inverting a GAN

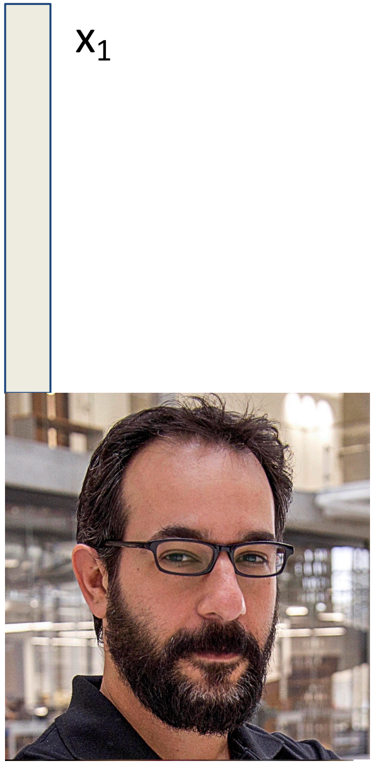


- Given a target image x_1 how do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 ?
- **Just define a loss $J(z) = || G(z) – x_1||$**
- **Do gradient descent on z (network weights fixed).**

Compressed sensing using generative models, Bora et al. ICML 2017.

Inverting a GAN

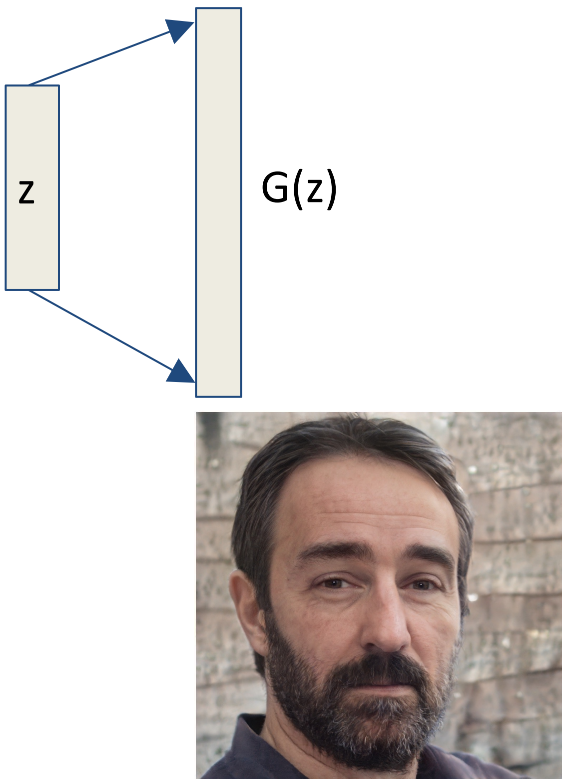


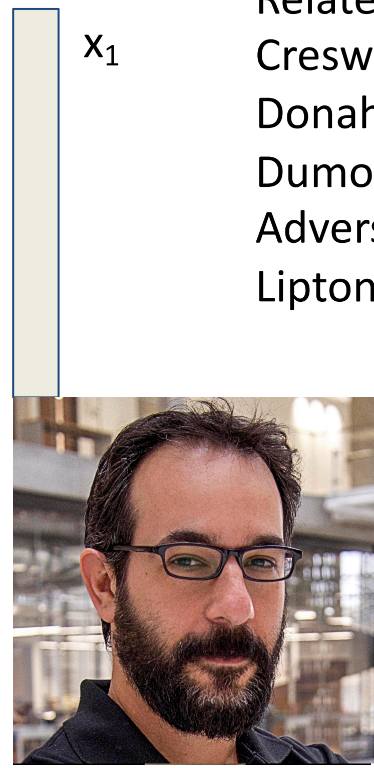


- Given a target image x_1 how do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 ?
- Just define a loss $J(z) = || G(z) - x_1||$
- Do gradient descent on z (network weights fixed).

Compressed sensing using generative models, Bora et al. ICML 2017.

Inverting a GAN



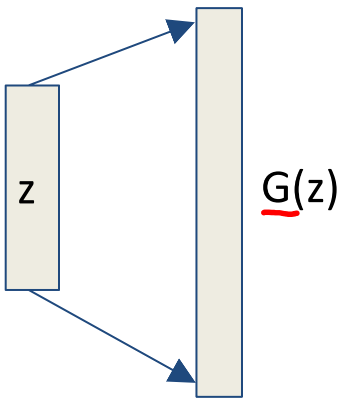


Related work :
Creswell and Bharath (2016)
Donahue, Krahenbuhl,Trevor 2016
Dumoulin et al.
Adversarially learned Inference
Lipton and Tripathi 2017

- Given a target image x_1 how do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 ?
- Just define a loss $J(z) = || G(z) - x_1 ||$
- Do gradient descent on z (network weights fixed).

Compressed sensing using generative models, Bora et al. ICML 2017.

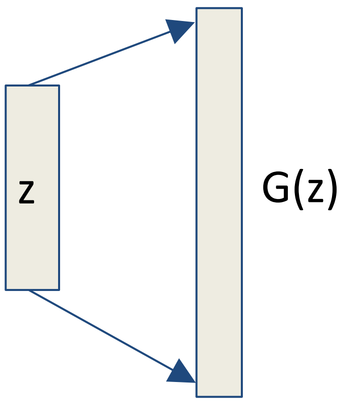
Recovery algorithm: Step 2: Inpainting



- Given a target image x_1 observe only some pixels.
- How do we invert the GAN now?

Compressed sensing using generative models, Bora et al. ICML 2017.

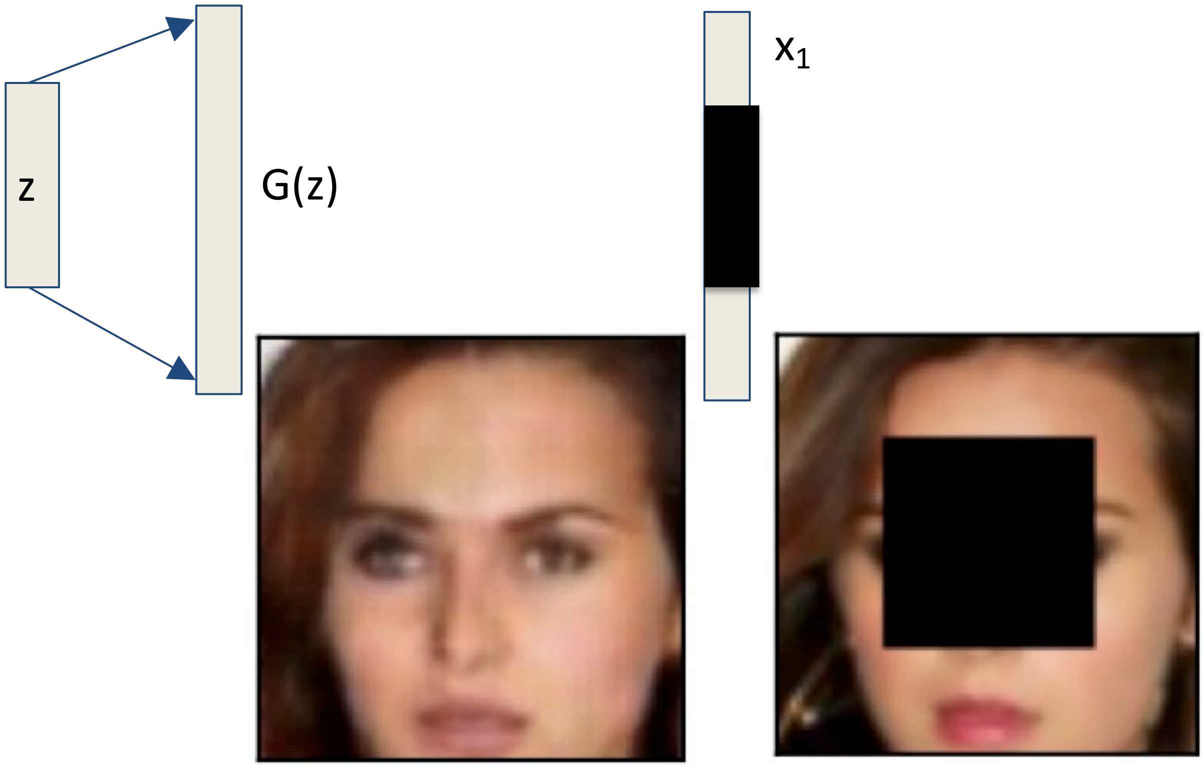
Recovery algorithm: Step 2: Inpainting



- Given a target image x_1 observe **only some pixels**.
- How do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 **on the observed pixels**?
- Just define a loss $J(z) = || A\, G(z) - A\, x_1||$
- Do gradient descent on z (network weights fixed).

Compressed sensing using generative models, Bora et al. ICML 2017.

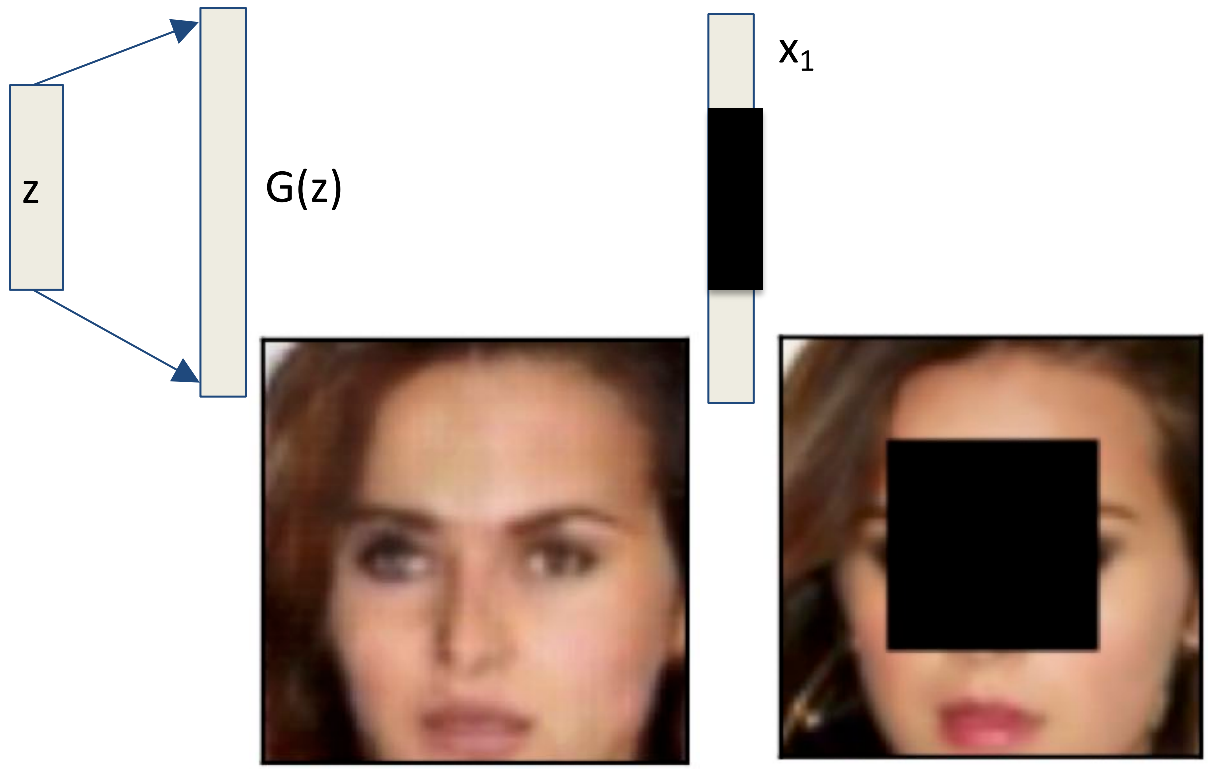
Recovery algorithm: Step 2: Inpainting



- Given a target image x_1 observe **only some pixels**.
- How do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 **on the observed pixels**?
- Just define a loss $J(z) = || A\,G(z) - A\,x_1||$
- Do gradient descent on z (network weights fixed).

Compressed sensing using generative models, Bora et al. ICML 2017.

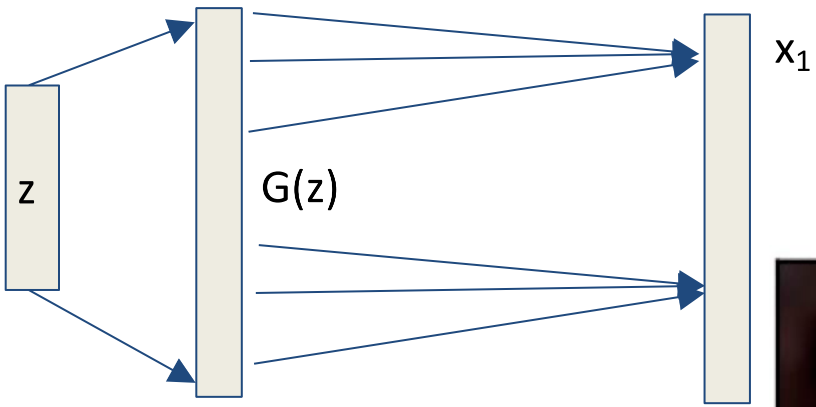
Recovery algorithm: Step 2: Inpainting



- Given a target image x_1 observe only some pixels.
- How do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 **on the observed pixels**?
- Just define a loss $J(z) = || A\,G(z) - A\, x_1||$
- Do gradient descent on z (network weights fixed).

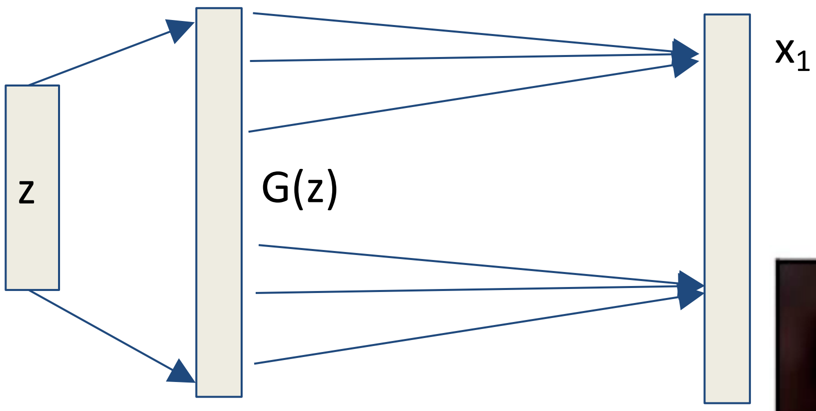
Compressed sensing using generative models, Bora et al. ICML 2017.

Recovery algorithm: Step 3: Super-resolution



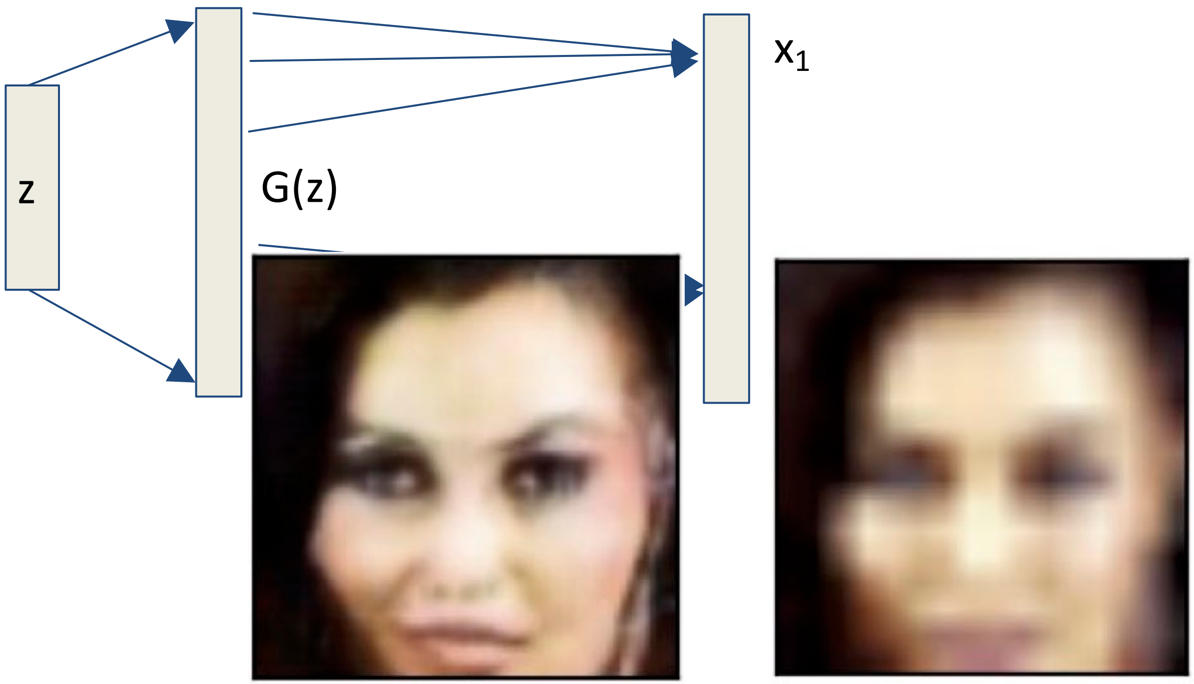
- Given a target image x_1 observe blurred pixels.
- How do we invert the GAN?

Recovery algorithm: Step 3: Super-resolution



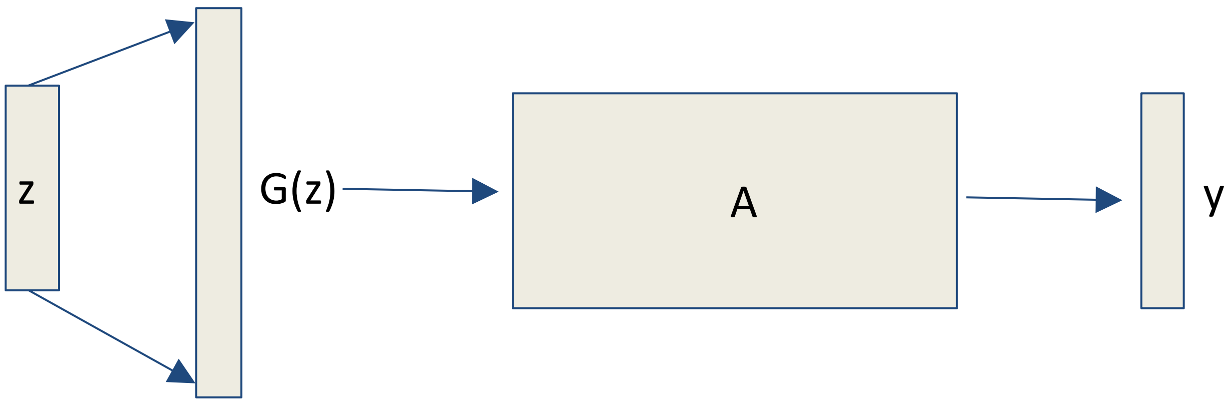
- Given a target image x_1 observe blurred pixels.
- How do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 **After it has been blurred**?
- Just define a loss $J(z) = || A\, G(z) - A\, x_1 ||$
- Do gradient descent on z (network weights fixed).

Recovery algorithm: Step 3: Super-resolution



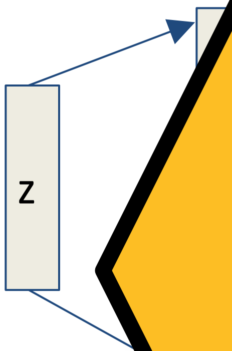
- Given a target image x_1 observe blurred pixels.
- How do we invert the GAN, i.e. find a z_1 such that $G(z_1)$ is very close to x_1 **After it has been blurred**?
- Just define a loss $J(z) = || A\,G(z) - A\, x_1 ||$
- Do gradient descent on z (network weights fixed).

Recovery from linear measurements



$$\min_{z \in \mathbb{R}^k} ||y - AG(z)||_2$$

Recovery from linear measurements



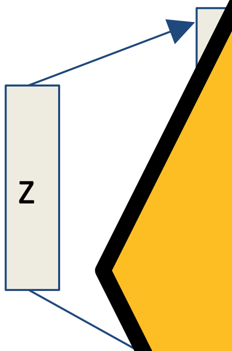
CSGM algorithm:
Do gradient descent in z space to satisfy measurements.

Obtain useful gradients
through the measurements using backprop.

$$\min_{z \in \mathbb{R}^k} ||y - AG(z)||_2$$

Compressed sensing using generative models, Bora et al. ICML 2017.

Recovery from linear measurements



CSGM algorithm:
Do gradient descent in z space to satisfy measurements.

Obtain useful gradients
through the measurements using backprop.

$$\min_{z \in \mathbb{R}^k} ||y - AG(z)||_2$$

Note: There are other methods for solving inverse problems
Supervised end-to-end inversion
CycleGAN, AmbientGAN and others.
Deep Learning Techniques for Inverse Problems in Imaging,
https://arxiv.org/pdf/2005.06001.pdf

Compressed sensing using generative models, Bora et al. ICML 2017.

Theory results

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$

Theory results

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$
- **Theorem 1:** If A is iid N(0, 1/m) with $m = O(kd \log n)$
- Then the reconstruction is close to optimal:

$$||G(\hat{z}) - x^*||_2 \leq C \min_z ||G(z) - x^*||$$

Theory results

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$
- **Theorem 1:** If A is iid N(0, 1/m) with $m = O(kd \log n)$
- Then the reconstruction is close to optimal:

$$||G(\hat{z}) - x^*||_2 \leq C \min_z ||G(z) - x^*|| + c||\eta||$$

- (Reconstruction accuracy relates to generator quality)
- **Thm2:** More general result: m = O(k log L) measurements for any L-Lipschitz function G(z)

Compressed sensing using generative models, Bora et al. ICML 2017.

- Let $y = Ax$
- Solve $\hat{z} =$ r
- **Theorem 1**
- Then the

$$||G($$

- (Reconstruction
- **Thm2:** More gen ny L-Lipschitz functio

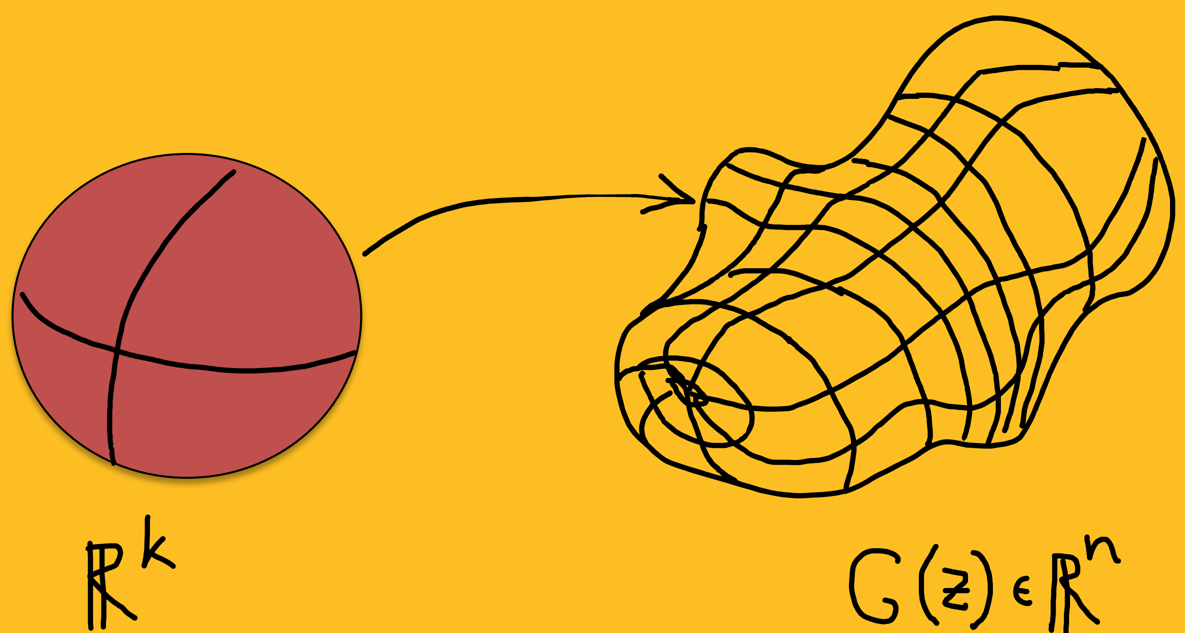
RIP property for a measurement matrix: all sparse vectors are far from the nullspace of measurement matrix

We define a **set restricted eigenvalue condition (S-REC)** that asks that **the differences of pairs of generated images is far from the nullspace.**

Key Lemma: Random matrices with m=k logL rows will satisfy S-REC whp.

- Let y
- Solve $\hat{z} =$
- Th
-

Key Lemma: Random matrices with
m=k logL rows will
satisfy S-REC whp.
How to bound metric entropy (aka log Covering
number) of generator range

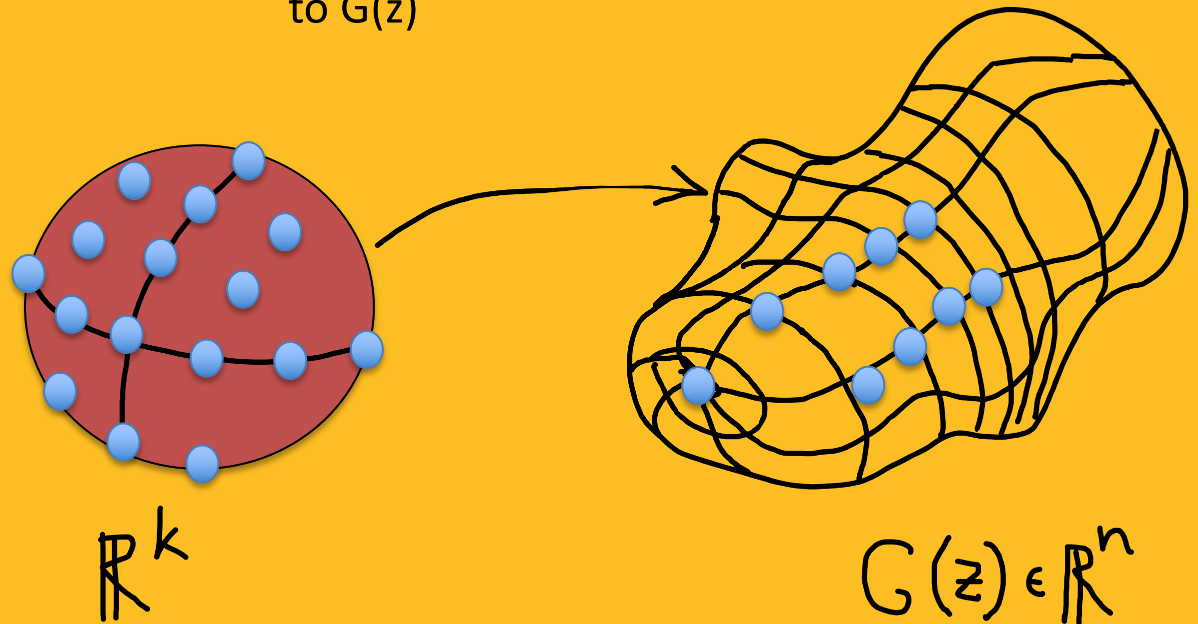


y

- Let y
- Solve $\hat{z} =$
- Th
-

Key Lemma: Random matrices with m=k logL rows will satisfy S-REC whp.
How to bound metric entropy (aka log Covering number) of generator range

Distance distortion L due to G(z)



y

Theory for Optimization

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$

  Open: How to do efficiently ?
  (under the right conditions)
- **Theorem 1:** If A is iid N(0, 1/m) with $m = O(kd \log n)$
- Then the reconstruction is close to optimal:

$$||G(\hat{z}) - x^*||_2 \leq C \min_z ||G(z) - x^*||$$

- (Reconstruction accuracy proportional to model accuracy)
- **Thm2:** More general result: m = O(k log L) measurements for any L-Lipschitz function G(z)

Compressed sensing using generative models, Bora et al. ICML 2017.

Theory for Optimization

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$

Open: How to do efficiently ?
(under the right conditions)

For generators with random iid weights, gradient descent provably solves this problem!
(Assuming each layer is logk factor bigger compared to previous one).

Hand and Voroninski
Global guarantees for enforcing deep priors by empirical risk (COLT 2018)

Leong, Hand, Voroninski
Phase Retrieval Under a Generative Prior (NeurIPS 2018)

Open problem: Unfortunately real generators have a contracting layer near the end.
Optimization for this (real) family of generators is open.

Talk Outline

Deep Generative models for inverse problems:

Compressed sensing using Generative models (CSGM)
(Bora et al. ICML 2017)
1. **Unsupervised** way to solve inverse problems using a deep generative model G(z)
2. CSGM Guarantees through Set Restricted Eigenvalue conditions.

Theory for Inverting deep generative models.
1.P. Hand and V. Voroninski, Global guarantees for enforcing deep priors by empirical risk, (COLT 2018)
2.Inverting Deep Generative models, One layer at a time
Q. Lei, A. Jalal, I. Dhillon, A.G.D, NeurIPS 2019
Daskalakis, Dhruv, and Zampetakis. "Constant-expansion suffices for compressed sensing with generative priors", 2021

A New Algorithmic idea: ILO: Intermediate Layer Optimization
Intermediate Layer Optimization for Inverse Problems using Deep Generative Models
ICML 2021, Deep-Inverse Workshop, NeurIPS 2020. Daras, Dean, Jalal, AD.
https://github.com/giannisdaras/ilo
https://arxiv.org/abs/2102.07364

New Results: ILO

ILO: Intermediate Layer Optimization for Inverse Problems using Deep Generative Models

ICML 2021,
Giannis Daras, Joseph Dean, Ajil Jalal, AD.
https://github.com/giannisdaras/ilo
https://arxiv.org/abs/2102.07364

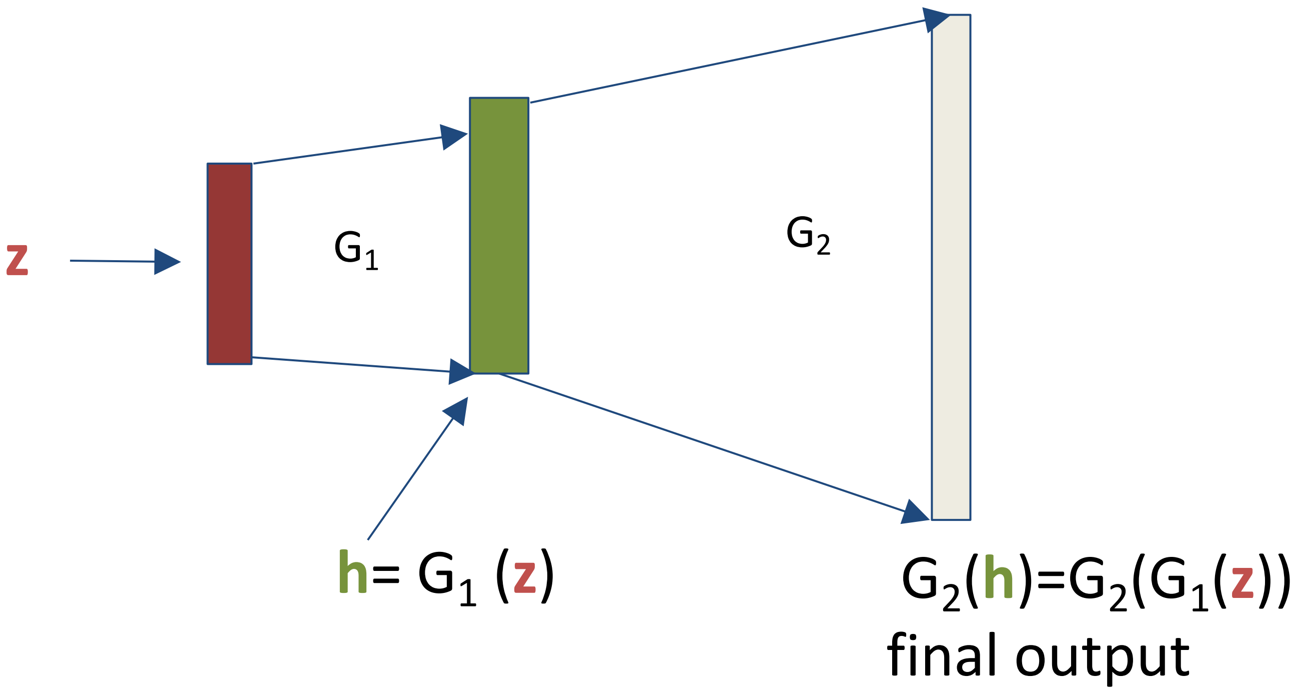
Two algorithmic innovations:

1. **Use Intermediate Layer Optimization (ILO)**
2. **Use LPIPS as a perceptual distance in addition to MSE**

And one more benefit:

-StyleGAN2 (1024x1024) versus 2017 DCGAN (64x64)

Intermediate Layer Optimization (ILO)



Consider a *nested* generator:

G= G_2 (G_1 (z))

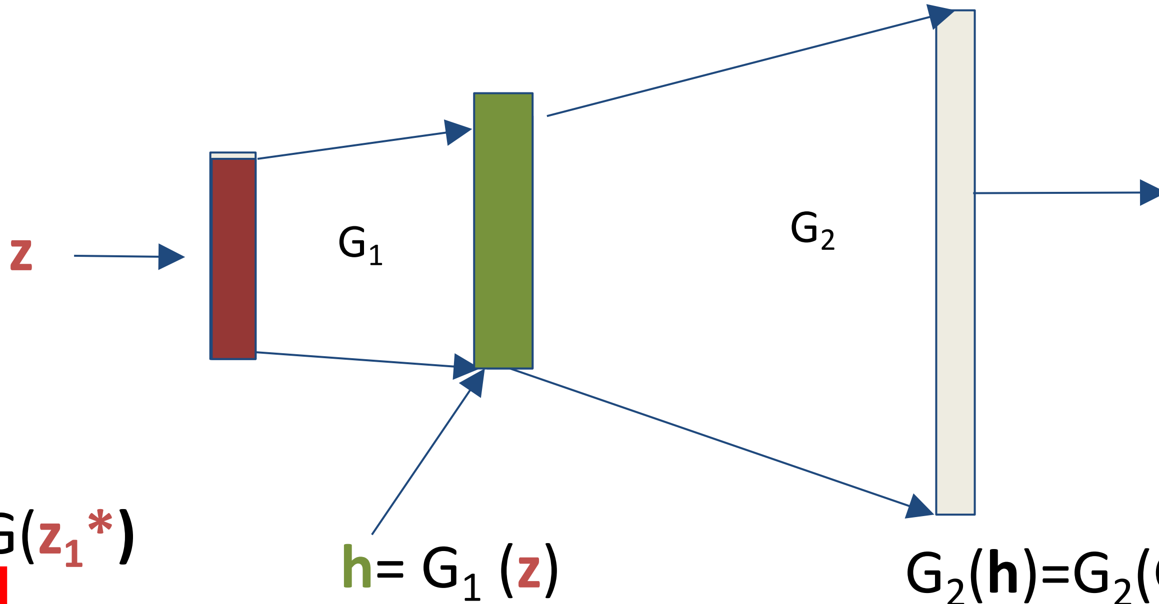
Step 1, Run CSGM: z_1^*= argmin_z $||G(z)\text{-}y||$= argmin_z $|| G_2 (G_1 (z)) - y||$

Step 2: After obtaining z_1^*, **optimize over h**:

h^*= $\text{argmin}_h || G_2 (h) - y ||$, starting with h_1=$G_1(z_1^*)$

Note that we may get non-realizable h vectors **hence we are expanding the range of the generator.**

ILO: Composition of Generators



$G_2(\mathbf{h})=G_2(G_1(\mathbf{z}))$
final output

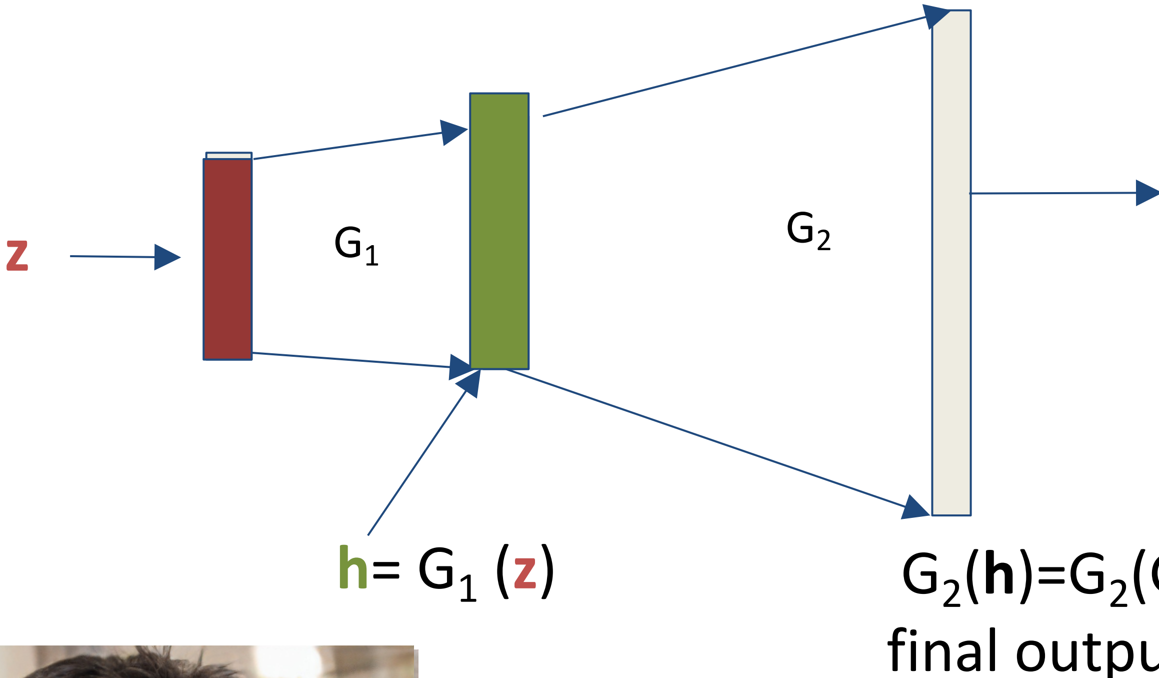
Target y

$G(z_1^*)$

G= G2 (G1 (z))

Step1: $z_1^* = \text{argmin } ||G(z)-y||$

ILO: Composition of Generators



Target y

G= G2 (G1 (z))

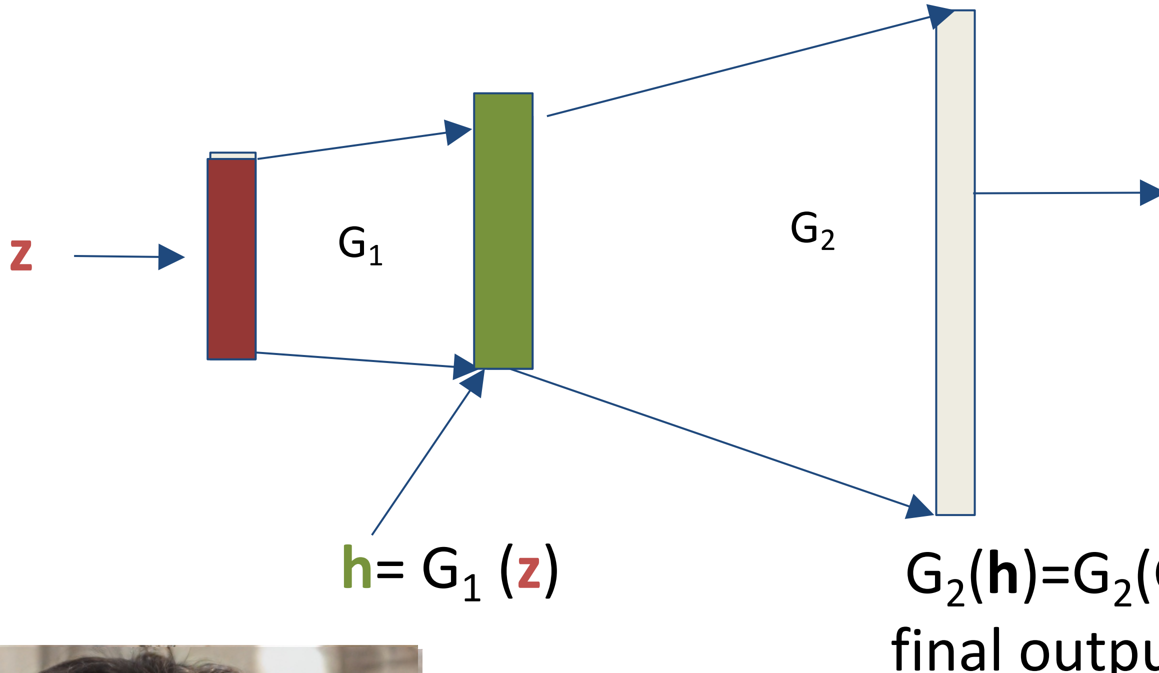
Step1: z_1^*= argmin_z ||G(z)-y|| (*normal CSGM*)

Step2: Min over **h**, starting with h_1= $G_1(z_1^*)$

G_2(**h***) , **h***= argmin_h|| G_2 (**h**) –y ||

Not a real picture

ILO in deeper layers expands the manifold too much



Target y

$G = G2(G1(z))$

Step1: $z_1^* = \text{argmin}_z \; ||G(z)-y||$ (*normal CSGM*)

Step2: Min over h, starting with $h_1 = G_1(z_1^*)$

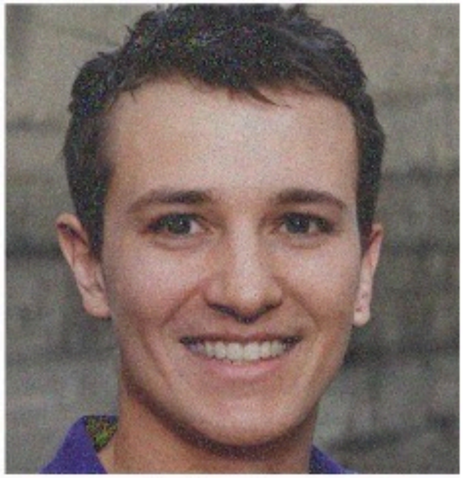
$G_2(h^*)$, $h^* = \text{argmin}_h || \; G_2(h) - y \; ||$

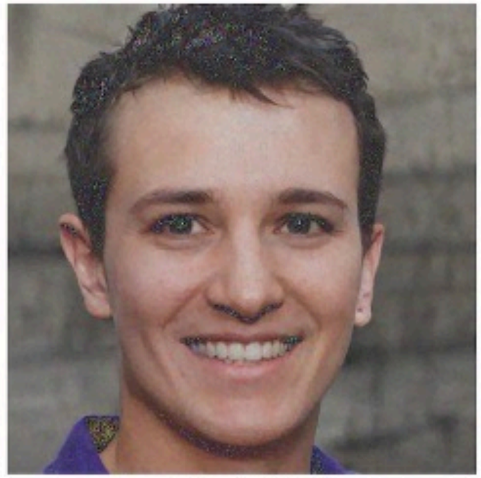
Intermediate Optimization in a deeper layer creates non-natural faces.

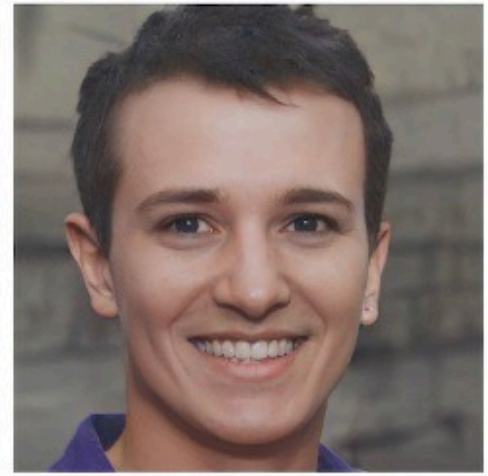
ILO results (inpainting)

Input: Corrupted images | ILO (inpainting) | Pulse (MSE) | Ground truth

ILO results (denoising)

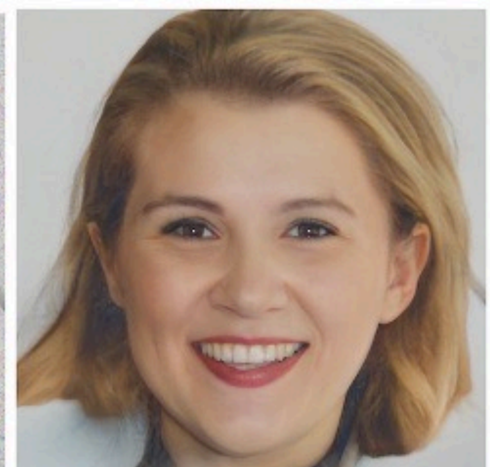

Noisy
(23.9dB)


BM3D
(27.6dB)


Ours
(30.0dB)


Noisy
(21.8dB)


BM3D
(24.4dB)


Ours
(27.1dB)

Super resolution with ILO

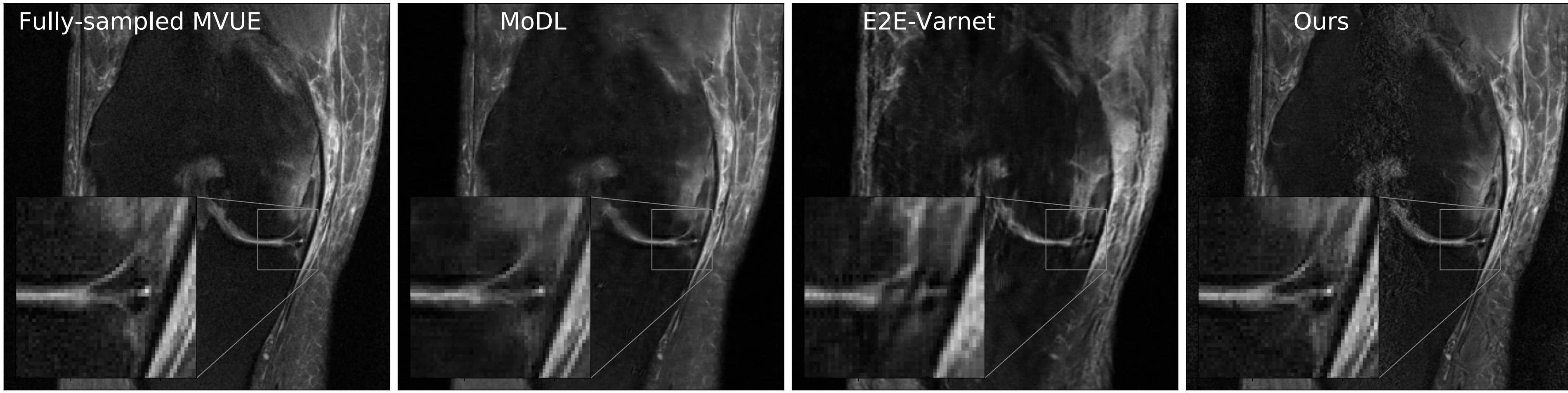
Input (LR 16x)

PULSE (previous SOTA)

ILO super-res. (Ours)

Ground truth

Applying Deep Generative models for MRI



Annotated Meniscus tear

4x Acceleration with diagnostically useful reconstructions is possible using deep-generative models.

Robust Compressed Sensing MRI with Deep Generative Priors (ICML'21)

We trained the first deep generative model for clinical MRI data.

Used Facebook FastMRI dataset

We **match** SOTA supervised Deep learning methods (in distribution)

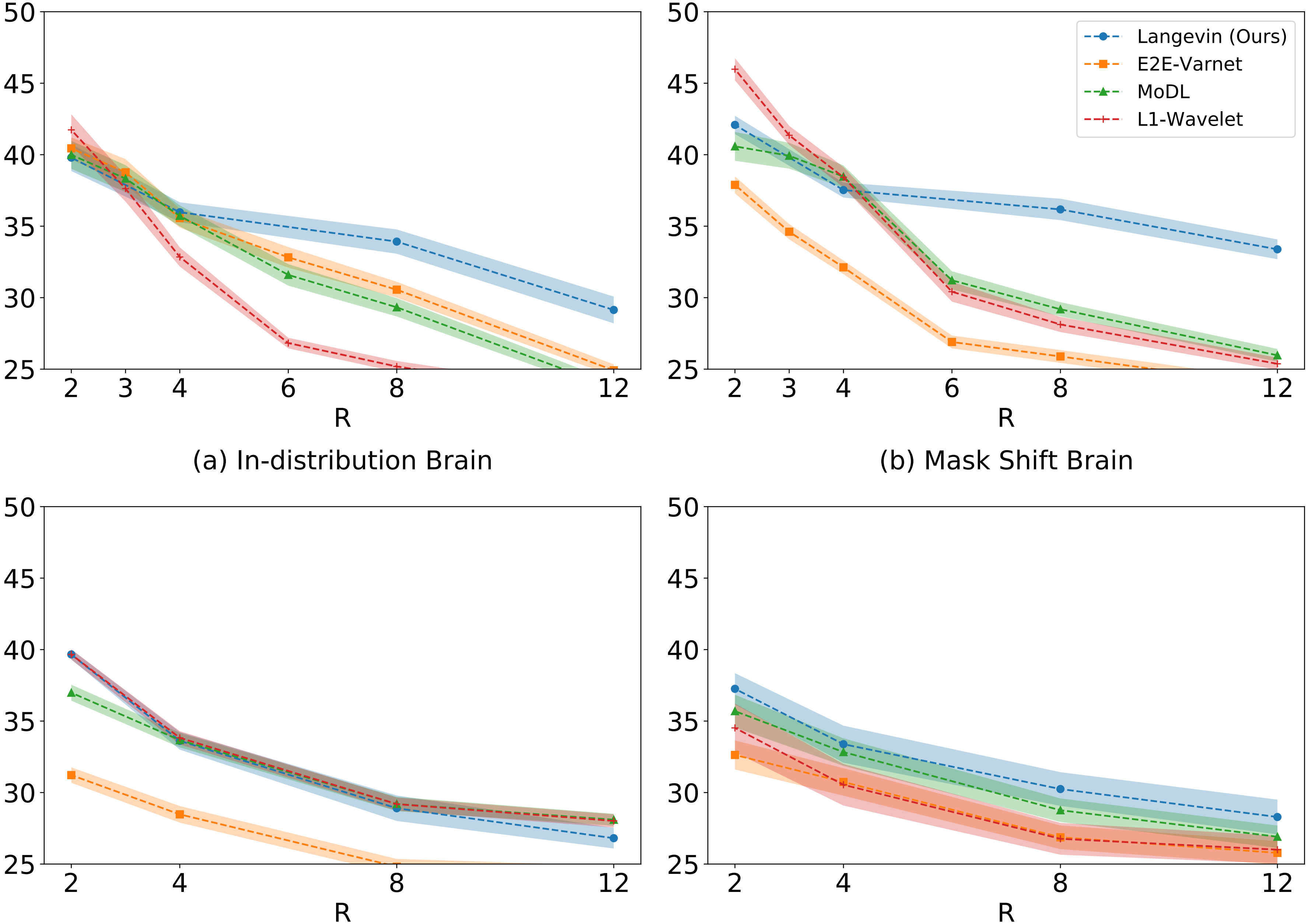
We **significantly outperform** SOTA supervised methods when MRI measurements change

We **mostly outperform** supervised methods under anatomy changes (train on Brains, reconstruct Knees)

Preliminary radiology evaluation: Our reconstructions are ranked as higher diagnostic quality in a blind evaluation by 3 experts. (or match supervised state of the art in other anatomies).

authors: Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, AGD, Jonathan I. Tamir

Robust Compressed Sensing MRI with Deep Generative Priors (ICML'21)

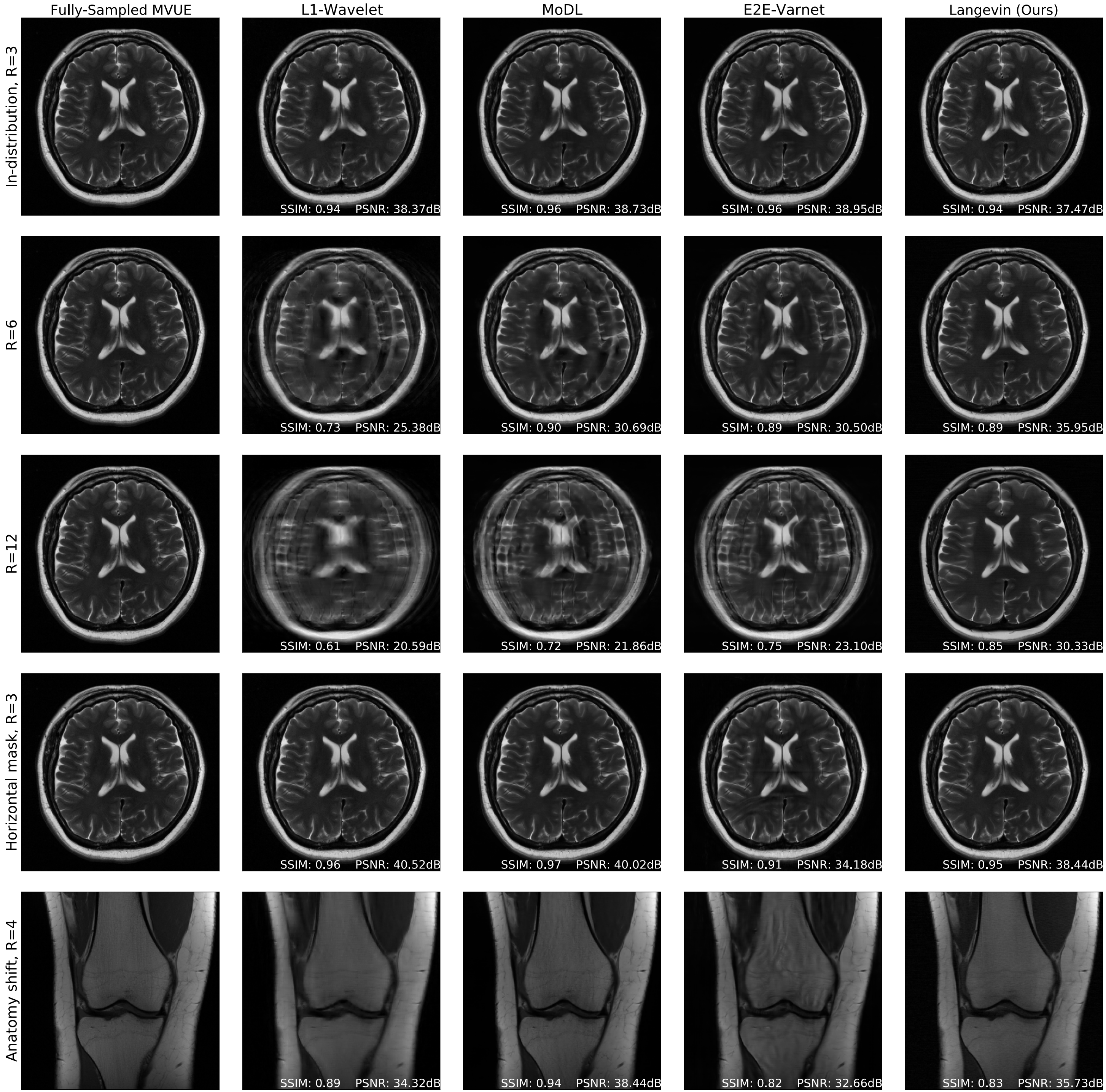


(a) In-distribution Brain

(b) Mask Shift Brain

(c) Abdomens

(d) Proton Density Knees


Fully-Sampled MVUE
L1-Wavelet
MoDL
E2E-Varnet
Langevin (Ours)
In-distribution, R=3
SSIM: 0.94 PSNR: 38.37dB
SSIM: 0.96 PSNR: 38.73dB
SSIM: 0.96 PSNR: 38.95dB
SSIM: 0.94 PSNR: 37.47dB
R=6
SSIM: 0.73 PSNR: 25.38dB
SSIM: 0.90 PSNR: 30.69dB
SSIM: 0.89 PSNR: 30.50dB
SSIM: 0.89 PSNR: 35.95dB
R=12
SSIM: 0.61 PSNR: 20.59dB
SSIM: 0.72 PSNR: 21.86dB
SSIM: 0.75 PSNR: 23.10dB
SSIM: 0.85 PSNR: 30.33dB
Horizontal mask, R=3
SSIM: 0.96 PSNR: 40.52dB
SSIM: 0.97 PSNR: 40.02dB
SSIM: 0.91 PSNR: 34.18dB
SSIM: 0.95 PSNR: 38.44dB
Anatomy shift, R=4
SSIM: 0.89 PSNR: 34.32dB
SSIM: 0.94 PSNR: 38.44dB
SSIM: 0.82 PSNR: 32.66dB
SSIM: 0.83 PSNR: 35.73dB

Blind Quality Assessment Results

- Two board-certified radiologists and a faculty member that uses neuroimaging in their research.
- 30 total blind quality assessment questions (3 anatomies x 10 scans). In each question, the experts were shown four images:
  - The fully-sampled reference image, explicitly marked as "Reference".
  - The results of three reconstruction algorithms at acceleration factor R=4: MoDL, ConvDecoder and our method. The order of the reconstructions was shuffled for each question.

	MoDL	ConvDec	Ours
Knee	1.87 (0.34)	2.97 (0.18)	**1.17** (0.45)
Abdomen	**1.87** (0.76)	**2.17** (0.93)	**1.97** (0.71)
Brain (in-dist)	**2.00** (0.82)	**2.07** (0.77)	**1.93** (0.85)

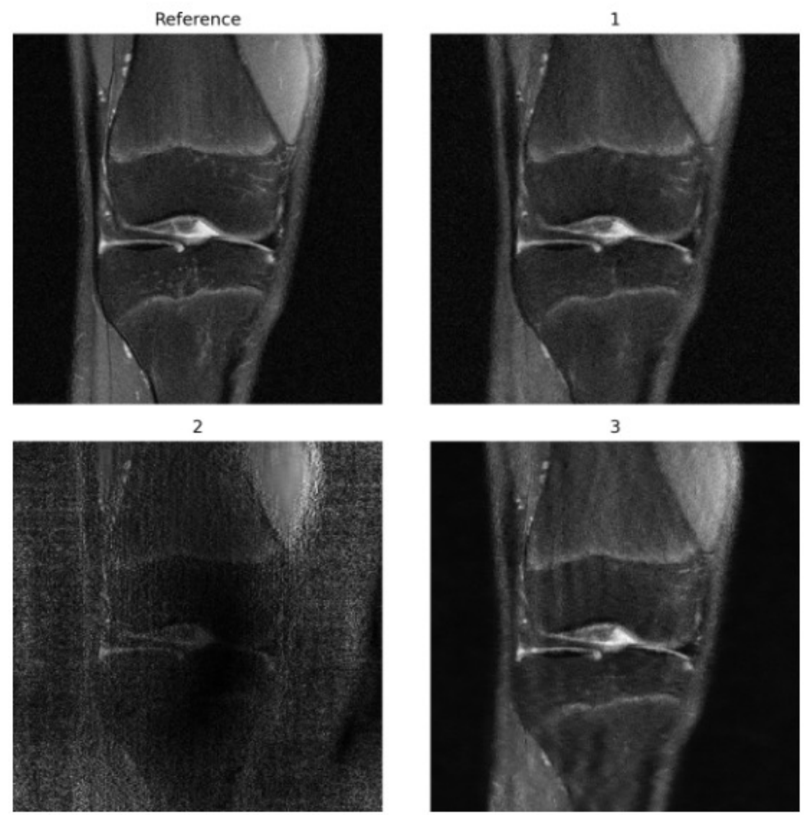
Average (std. dev.) ranking (N = 30) of each method on each anatomy. Lower is better.

	Ours vs. MoDL	Ours vs. ConvDec
Knee	1.53e-10	2.77e-6
Abdomen	0.610	0.340
Brain	0.767	0.550

Pairwise p-values for the hypothesis that rankings are significantly different.

Blind Quality Assessment survey example

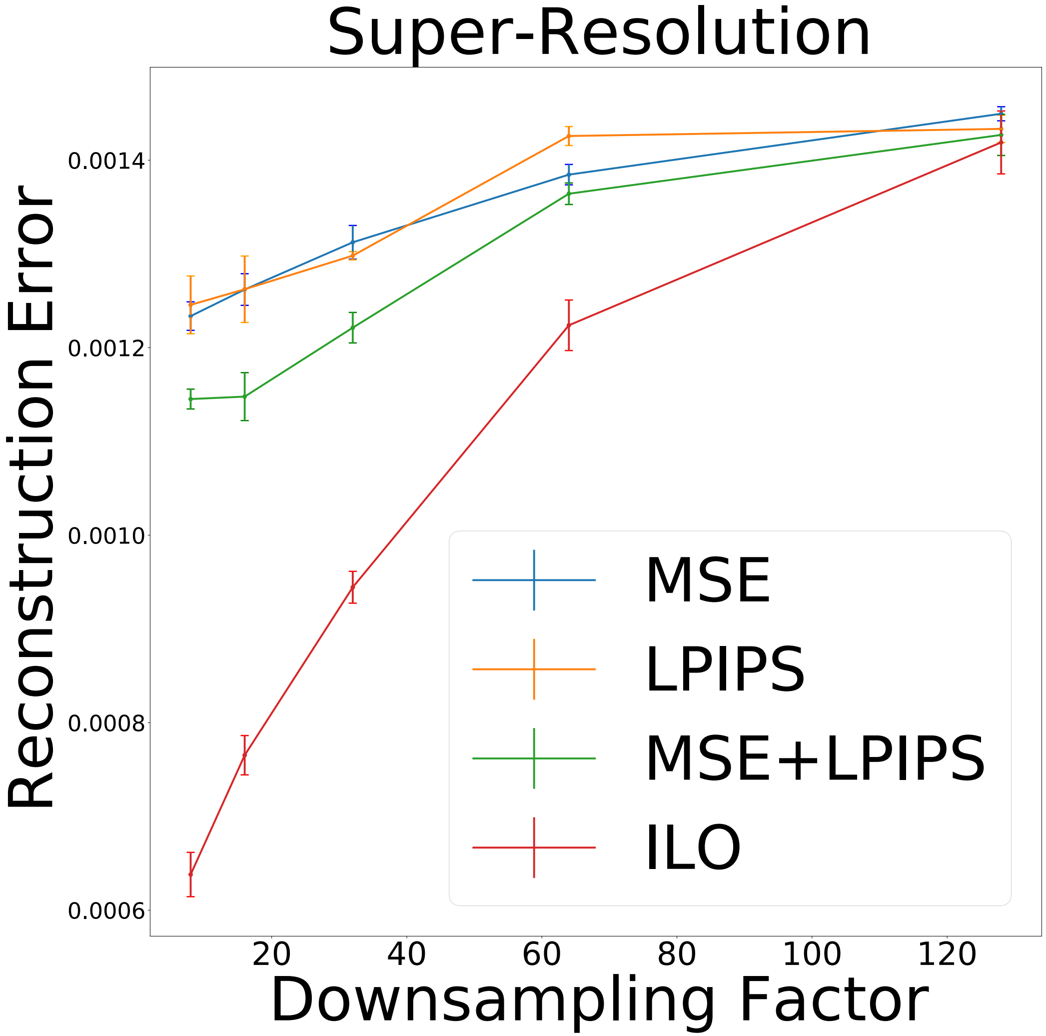
Consider the images below containing a **reference, fully-sampled** (top-left) reconstruction and the **three** reconstructions to be evaluated, labeled appropriately.



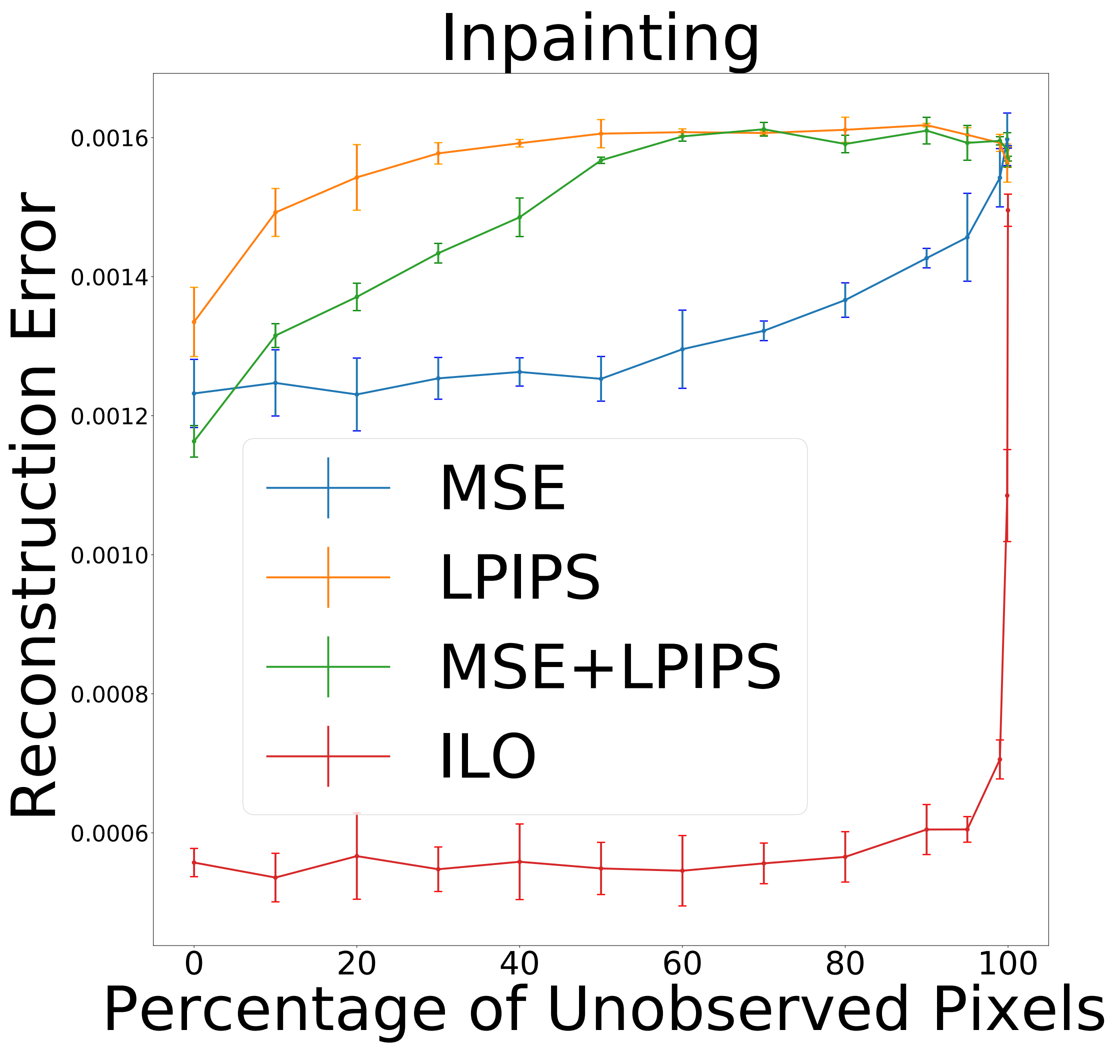
Please use the buttons below to choose which of the three reconstructions is perceptually best, second-best and third-best respectively.

	1	2	3
Best reconstruction	○	○	○
Second-best reconstruction	○	○	○
Third-best reconstruction	○	○	○

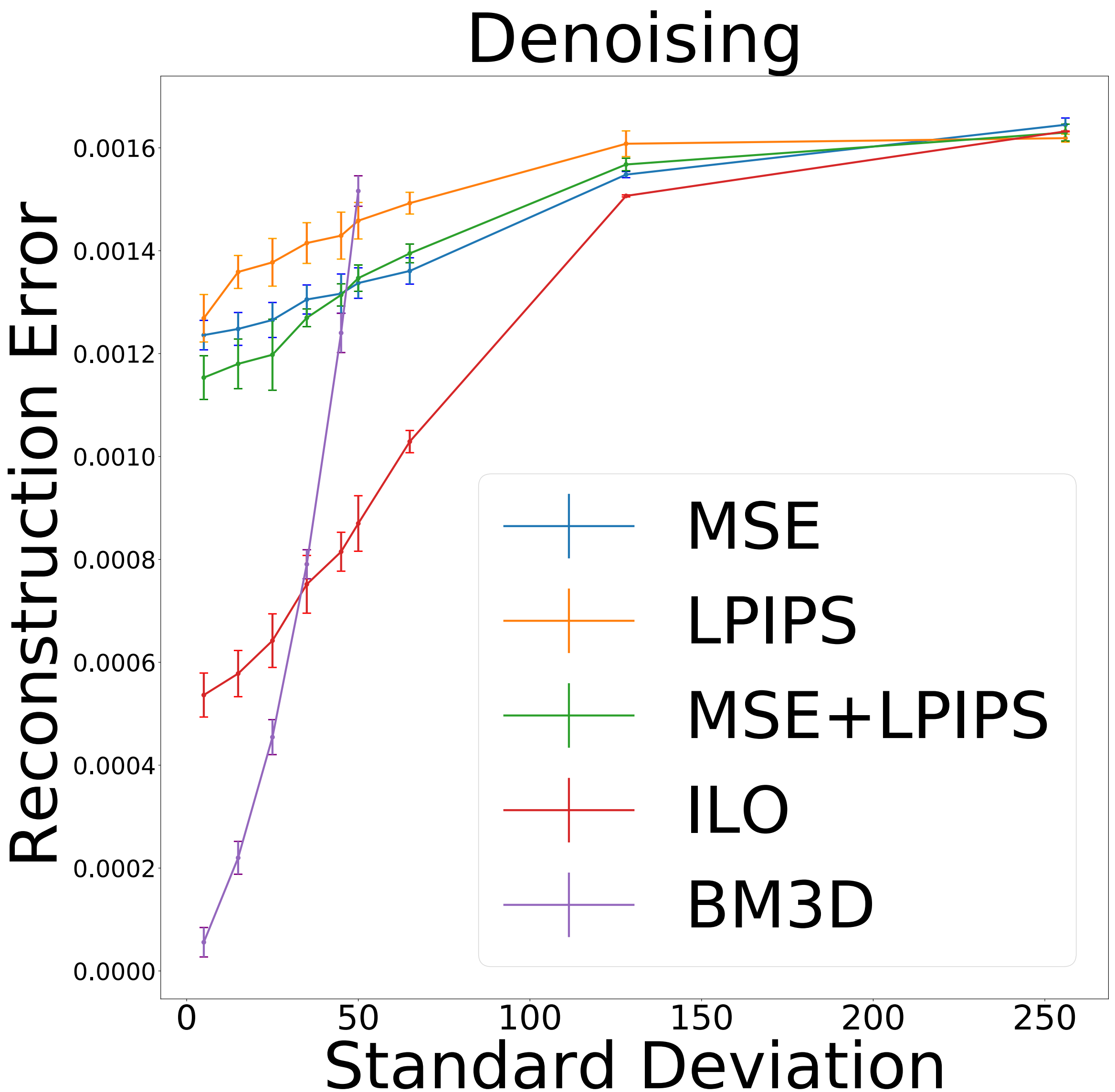
Super-resolution: ILO versus PULSE



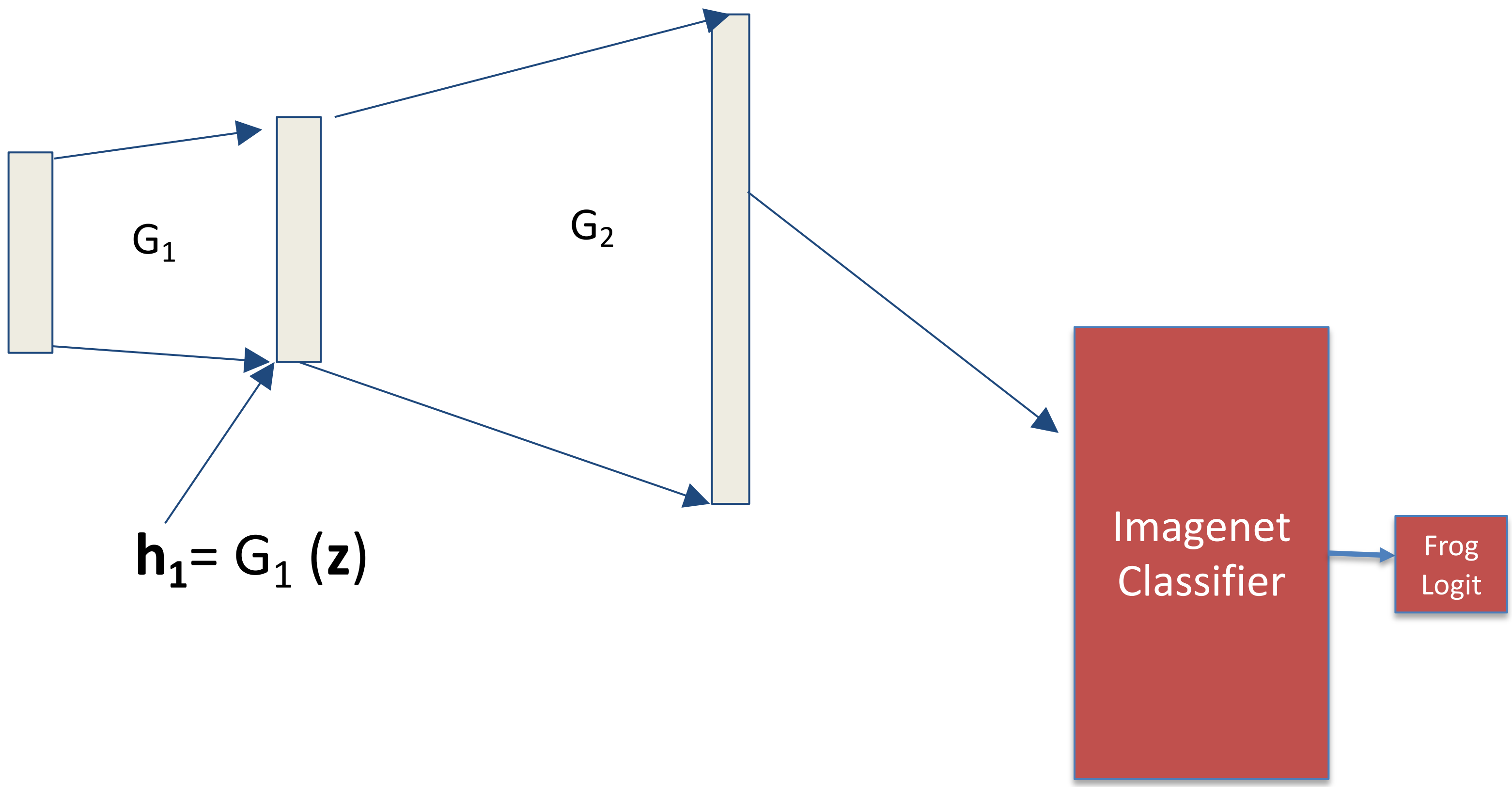
Inpainting: CSGM vs ILO



Gaussian denoising: CSGM vs ILO vs BM3D



How to make frog-people

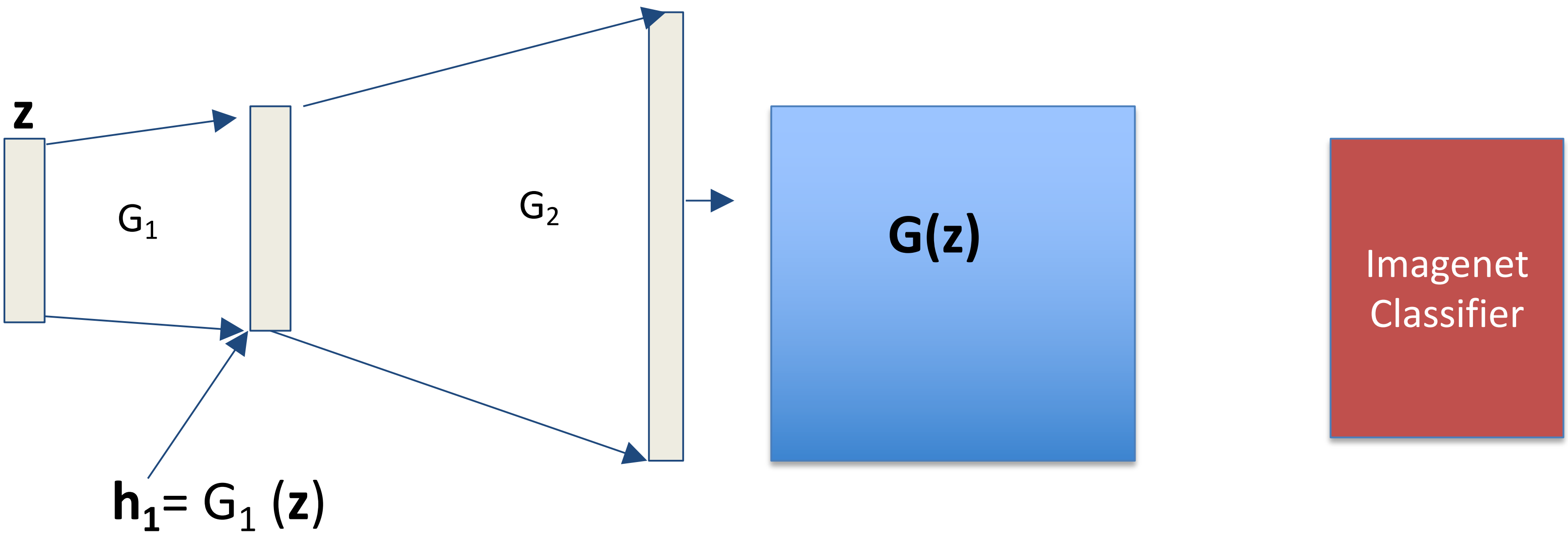


max **Frogness(**G(z))

Frog human

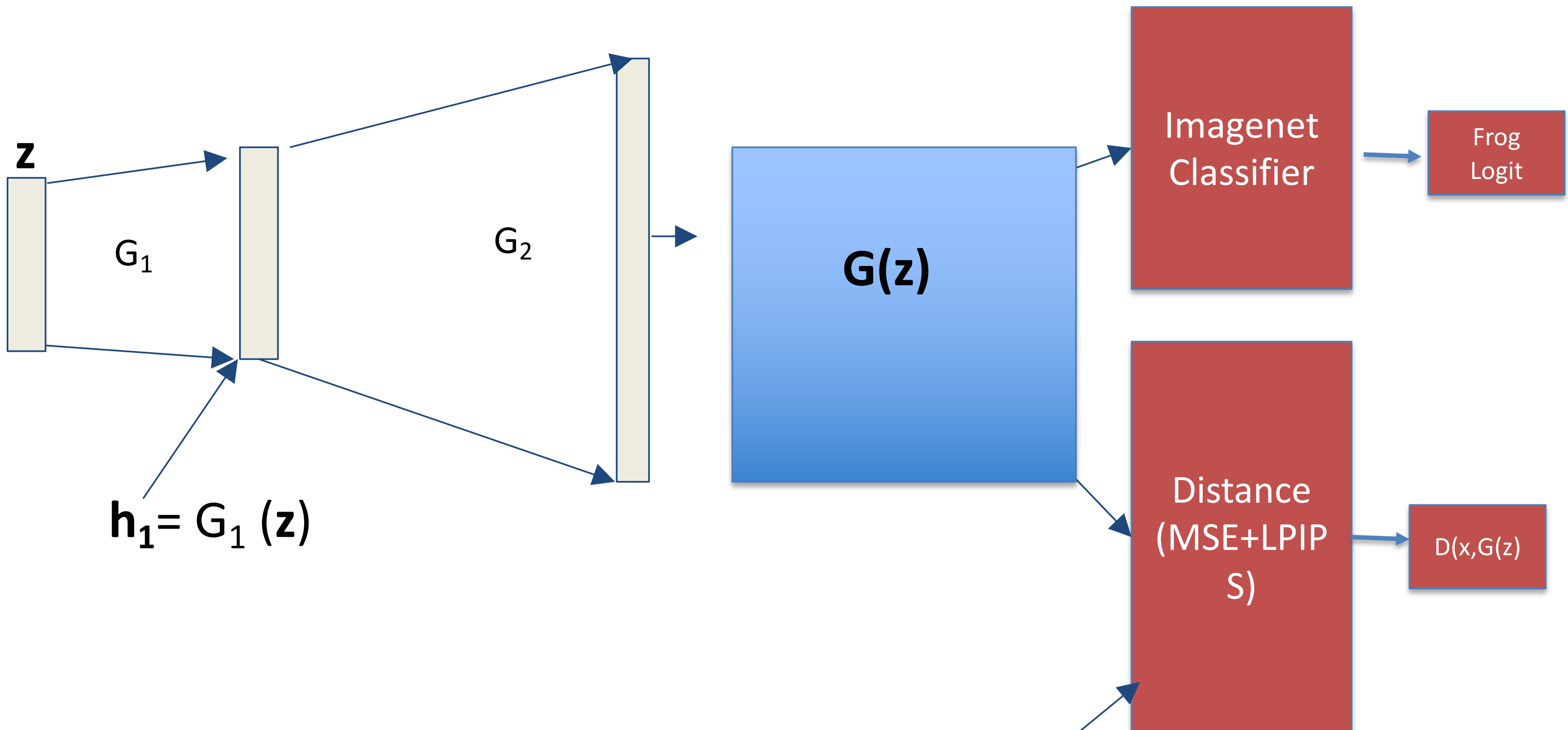
Goldfish human

3. Turning your friends into Frog-people




Friend x1

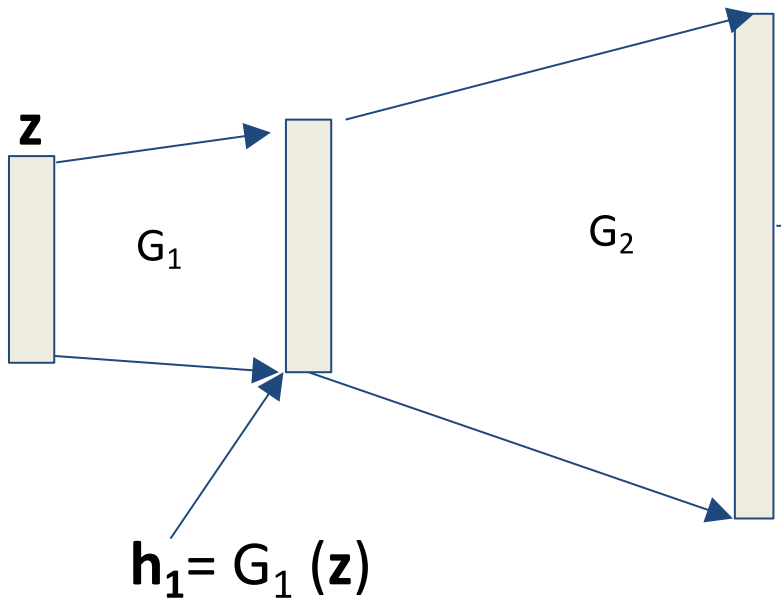
3. Turning your friends into Frog-people

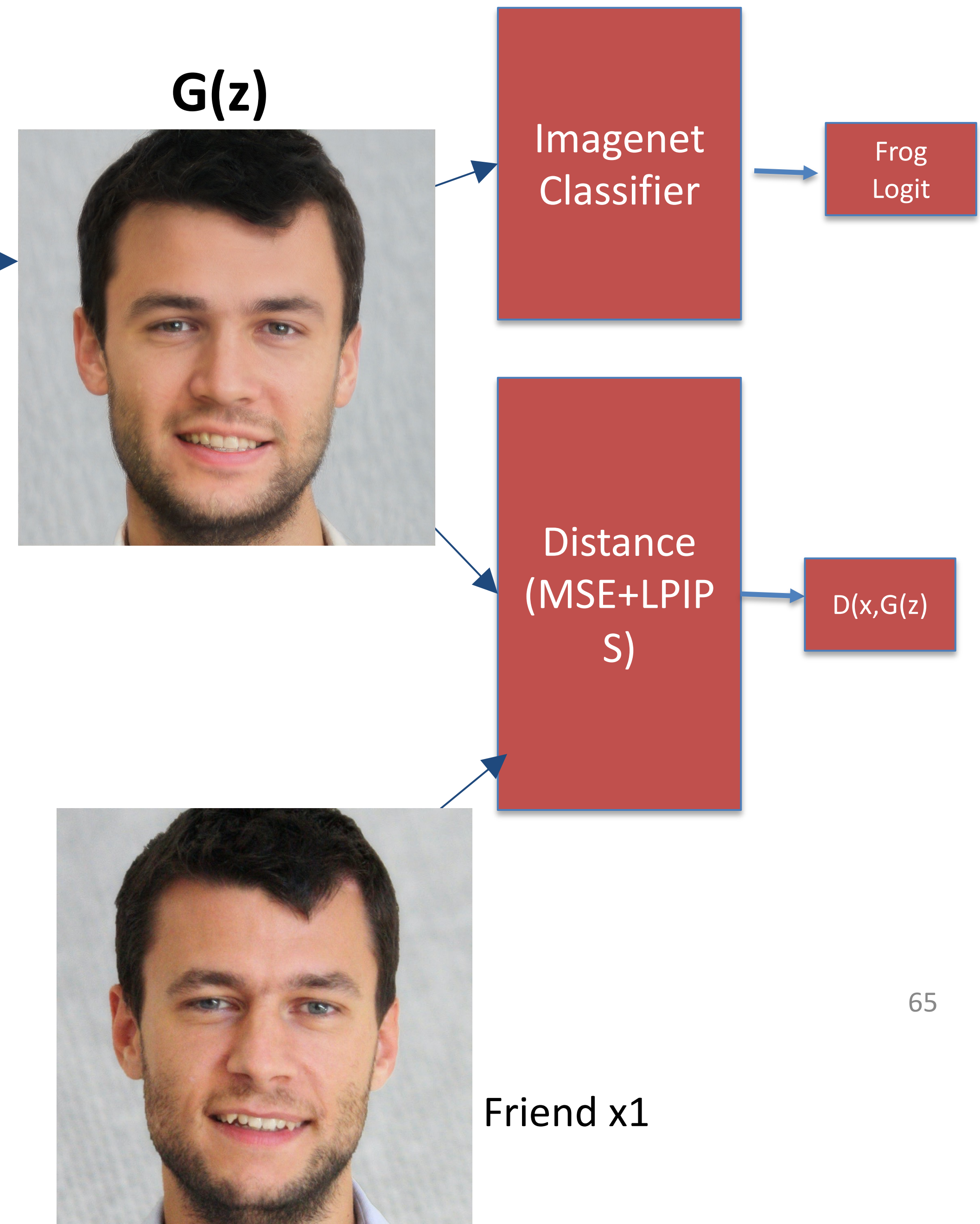


argmin **Frogness(**G(z)) +
λ Dist[x1, G(z)]

Friend x1

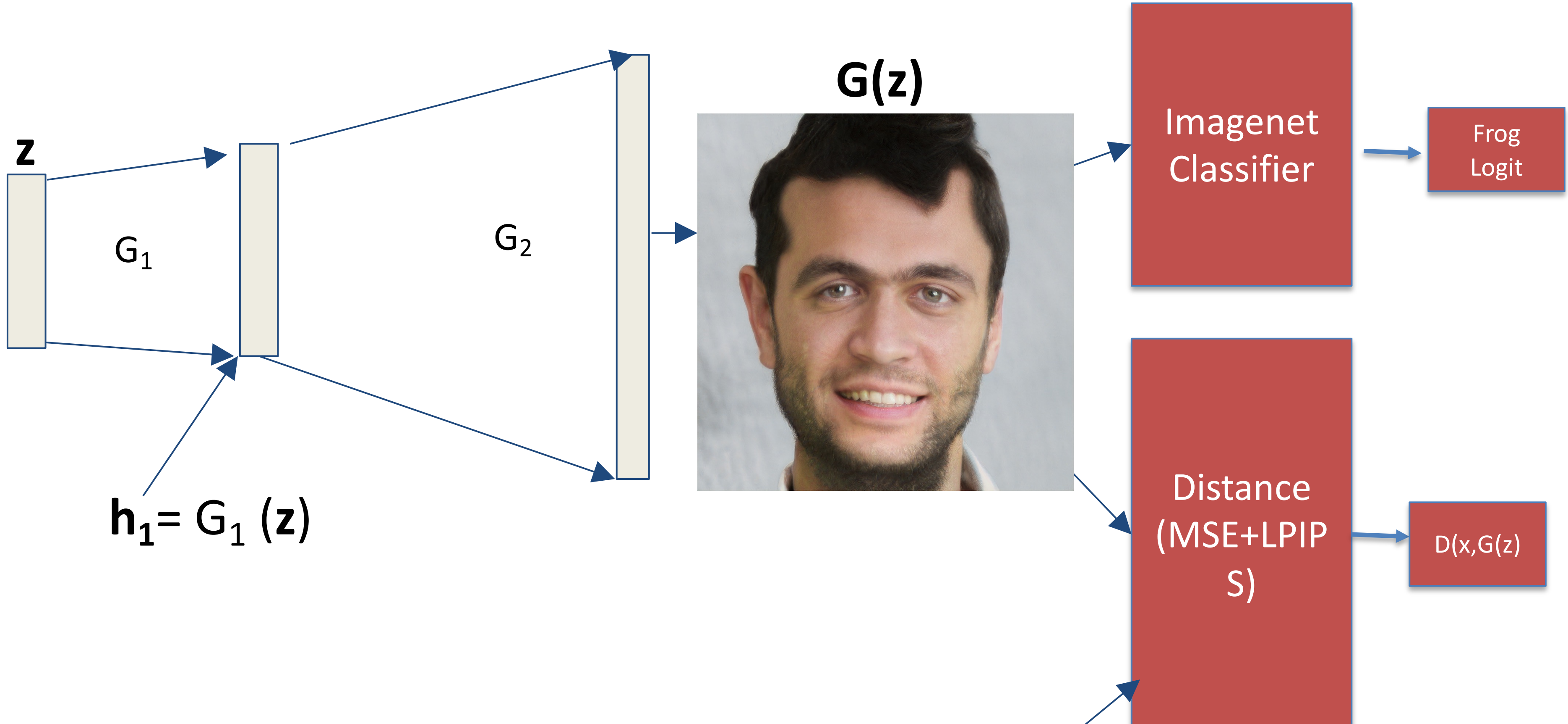
3. Turning your friends into Frog-people



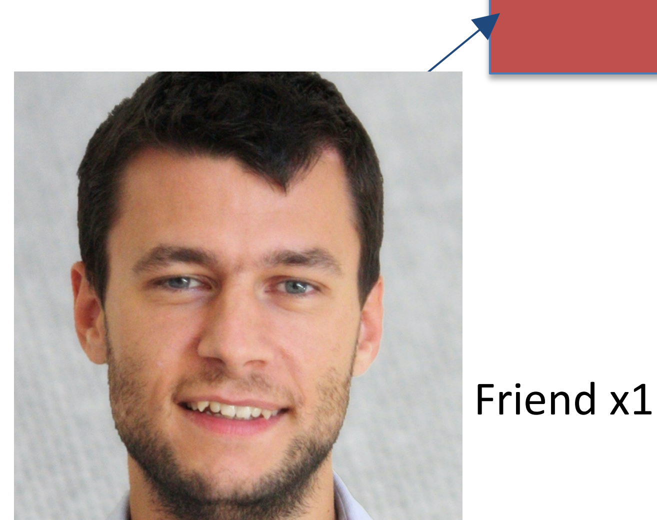


argmin 0***Frogness(**G(z)) +
λ Dist[x1, G(z)]

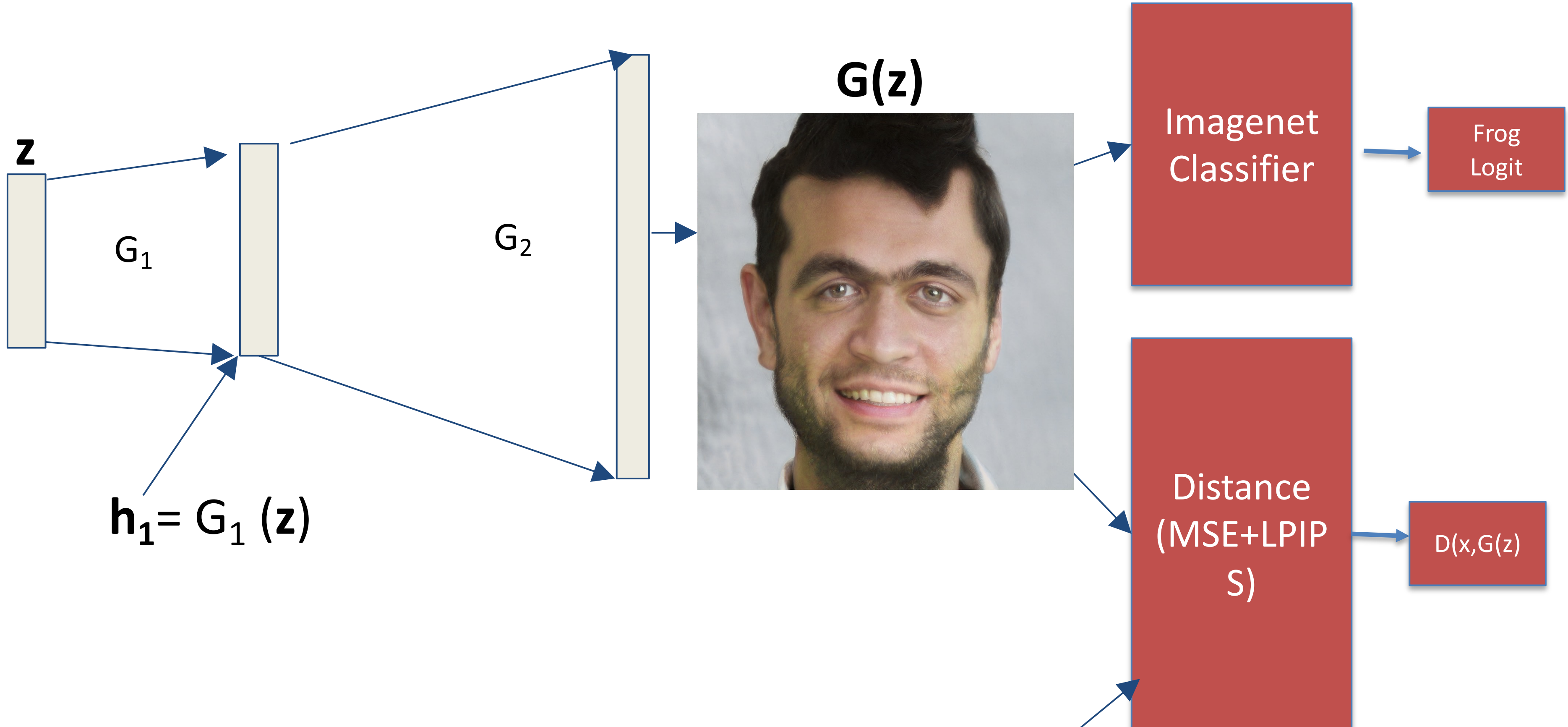
3. Turning your friends into Frog-people



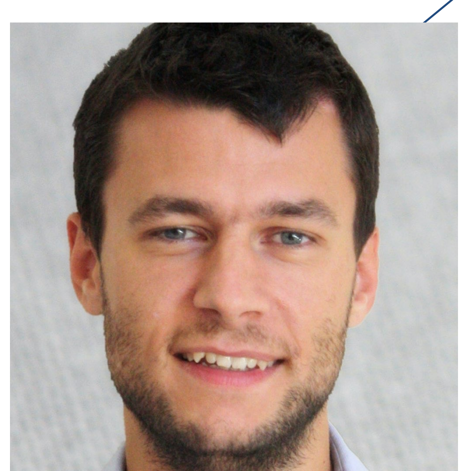
argmin 1***Frogness(**G(z)) +
λ Dist[x1, G(z)]



3. Turning your friends into Frog-people

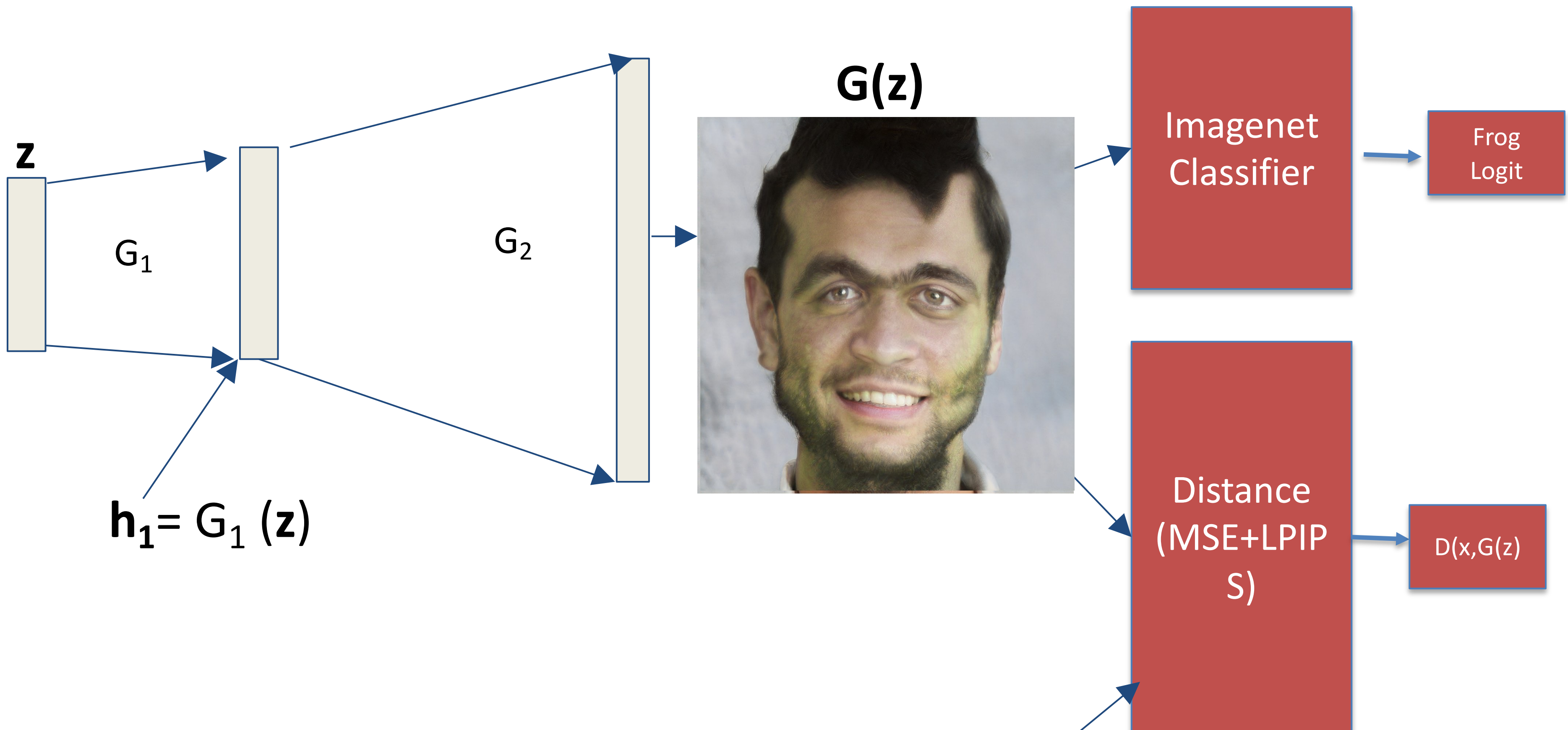


argmin 1.5* **Frogness(**G(z)) +
λ Dist[x1, G(z)]

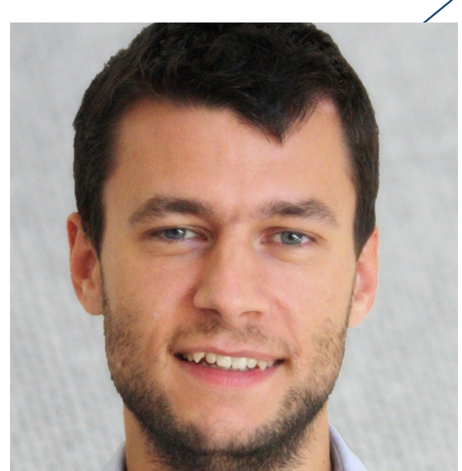


Friend x1

3. Turning your friends into Frog-people

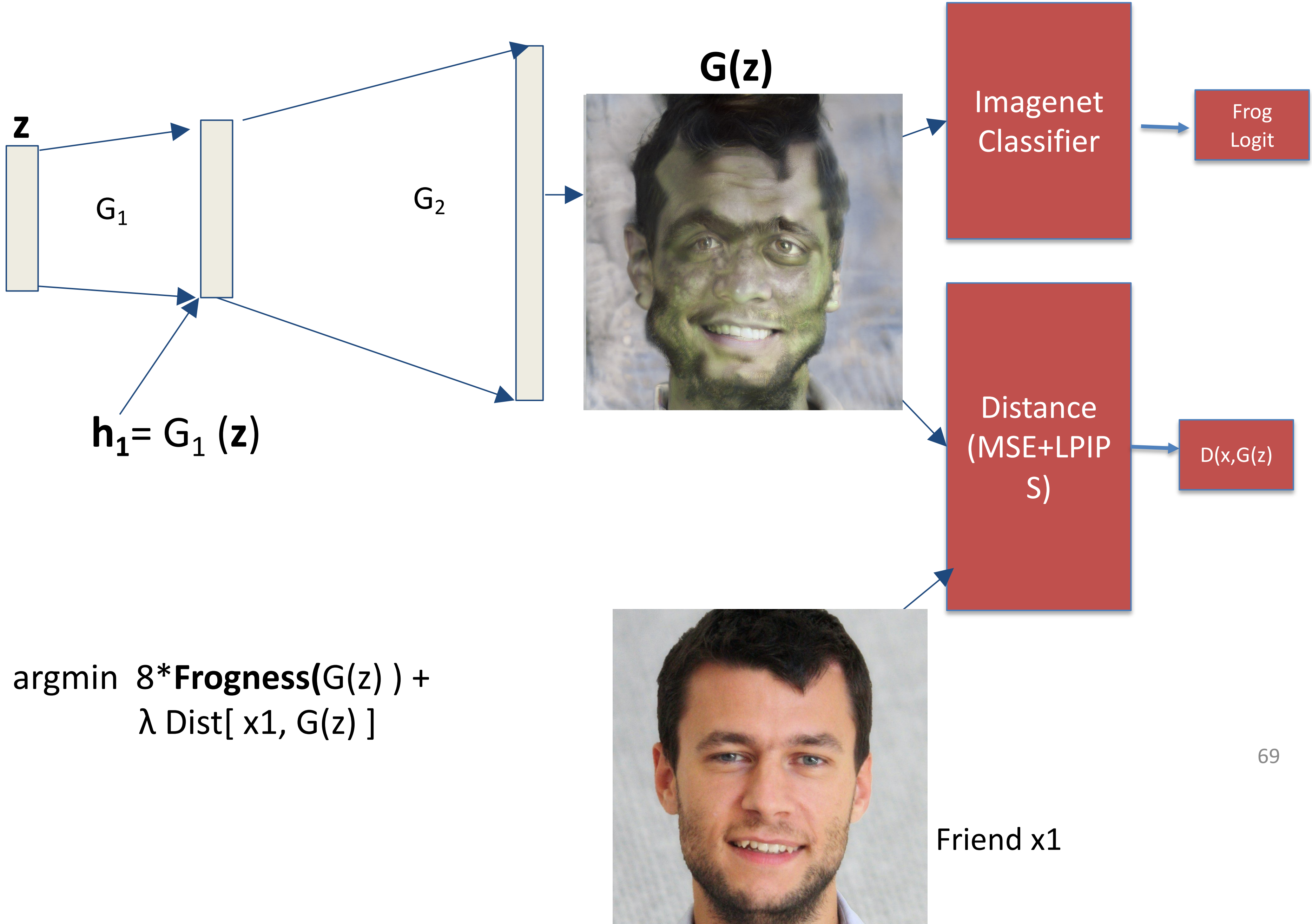


argmin 3***Frogness(**G(z)) +
λ Dist[x1, G(z)]



Friend x1

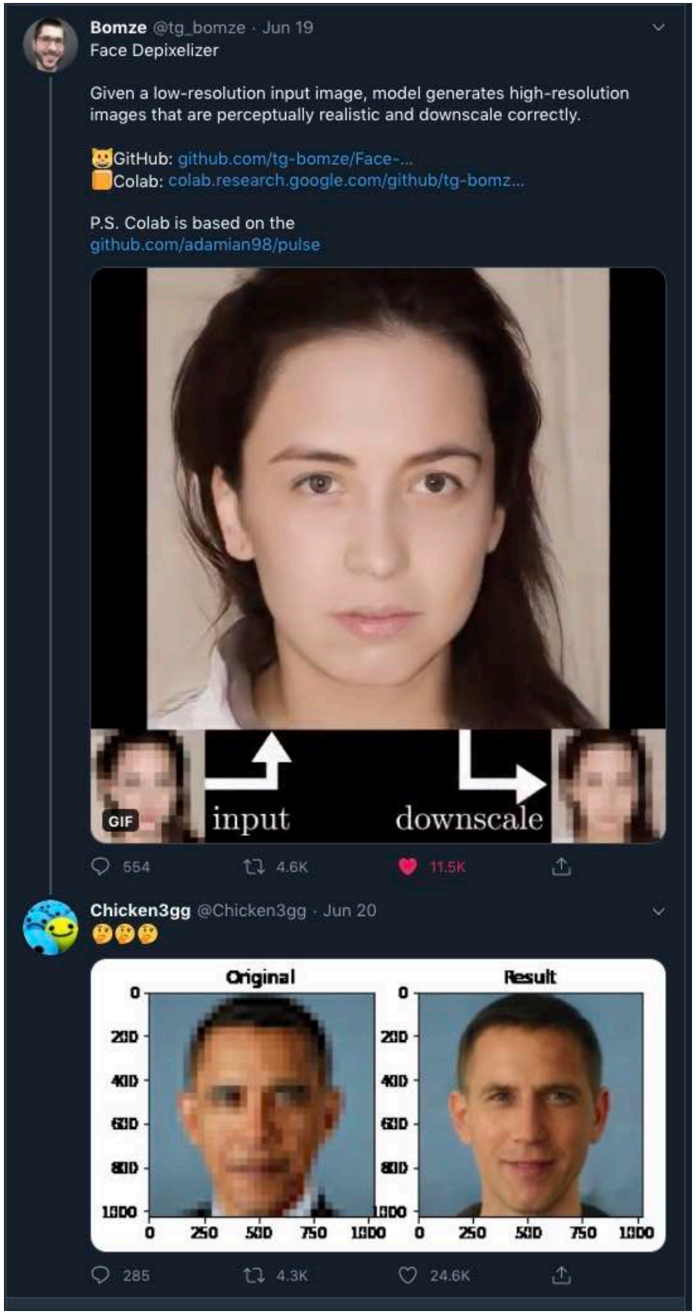
3. Turning your friends into Frog-people



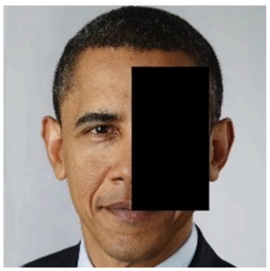
argmin 8***Frogness(**G(z)) +
λ Dist[x1, G(z)]

Not Fun part: Bias in inverse problems

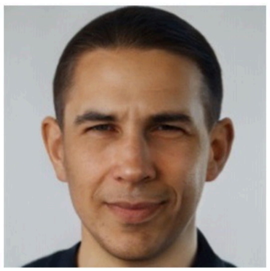
Bias in inverse problems



ILO results for Obama Inpainting

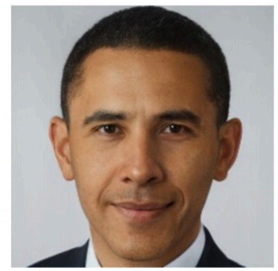


Observation



PULSE (MSE)

LPIPS



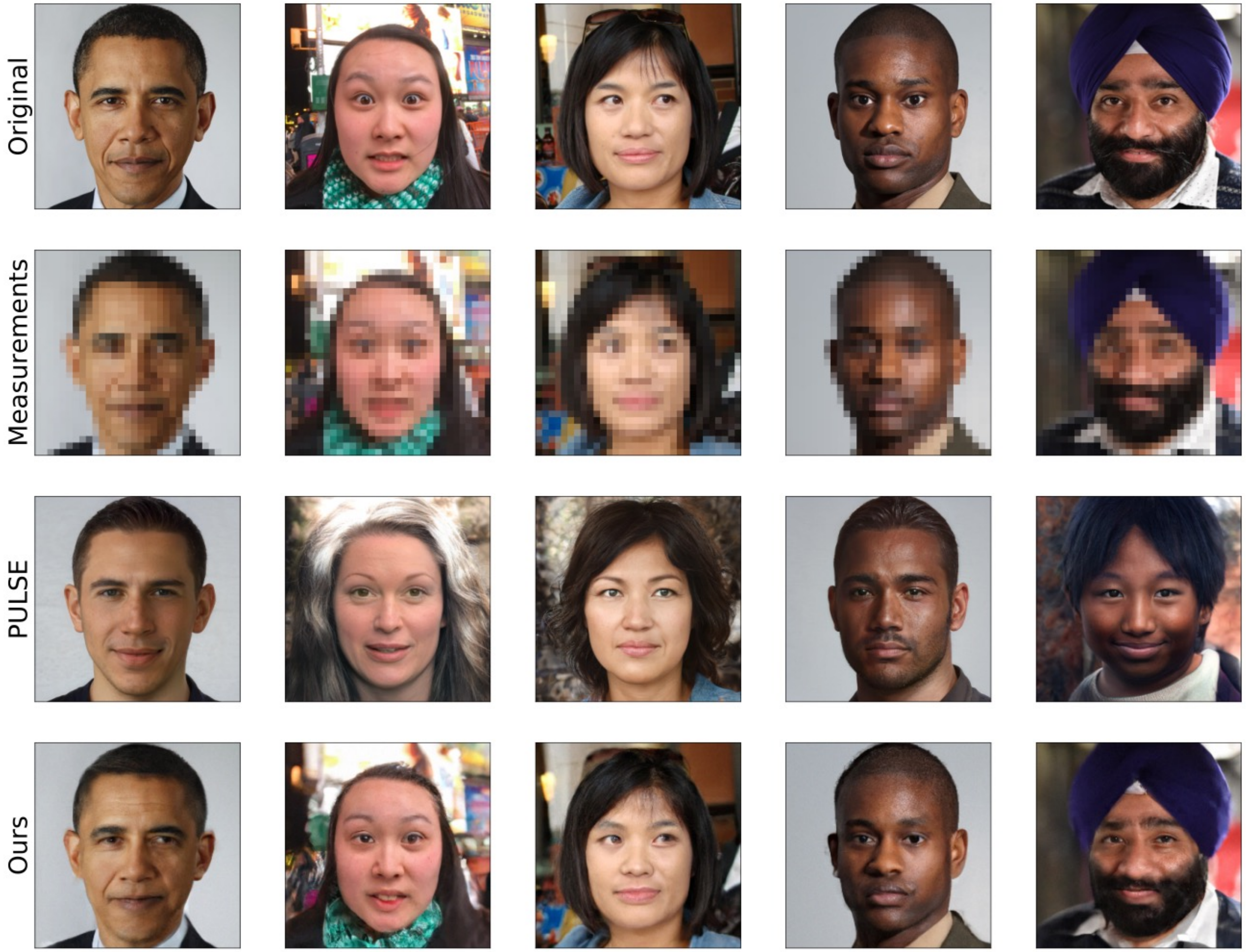
LPIPS+ MSE

ILO with LPIPS+ MSE

Real image

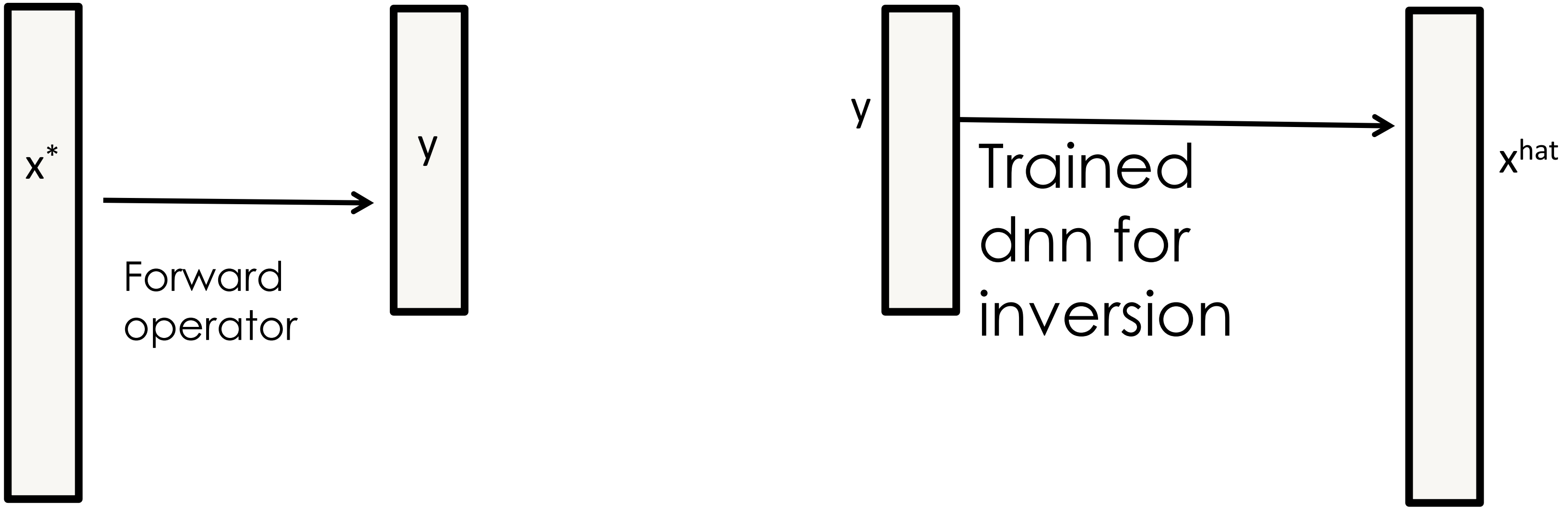
Intermediate Layer Optimization for Inverse Problems using Deep Generative Models
Deep-Inverse Workshop, NeurIPS 2020.
Joseph Dean, Giannis Daras, AD.

Fairness in inverse problems



[Fairness for Image Generation with Uncertain Sensitive Attributes](). A. Jalal et al. ICML 2021

End to end Approach



- One idea: end-to-end inversion using DNNs. (100s of papers propose this)
- Create many (x*,y) pairs using a simulator. Train a network to go from y to x*.
- Key issues: how to get a good matched dataset.
- How to design and train the inversion net (e.g. U-Nets, ADMM or unrolling methods)
- End-to-end methods are very fragile to uncertainty in forward operator or x statistics. (PNAS 2021, ICML 2021, Our just submitted paper)

Supervised End2End methods are Brittle

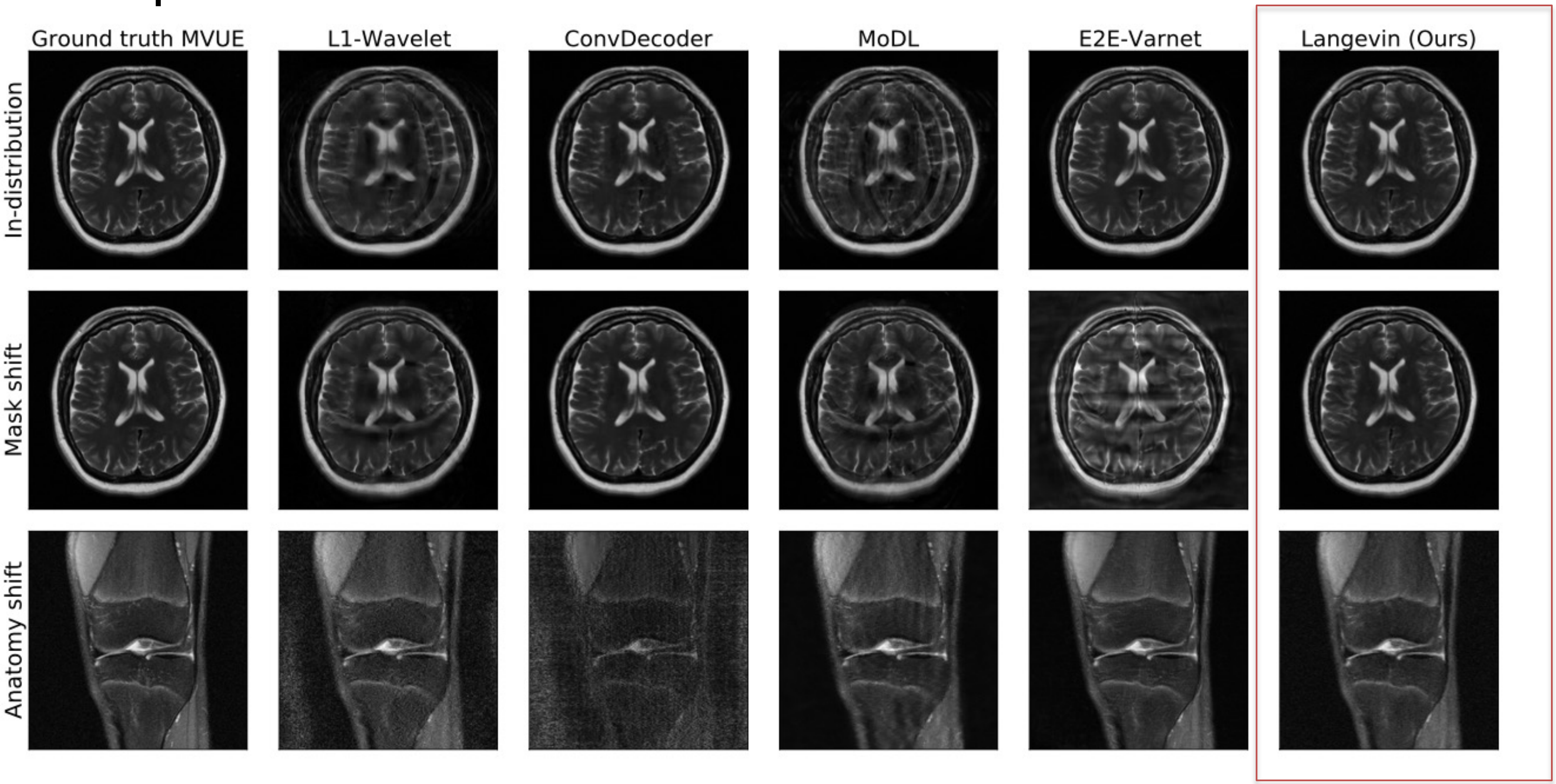
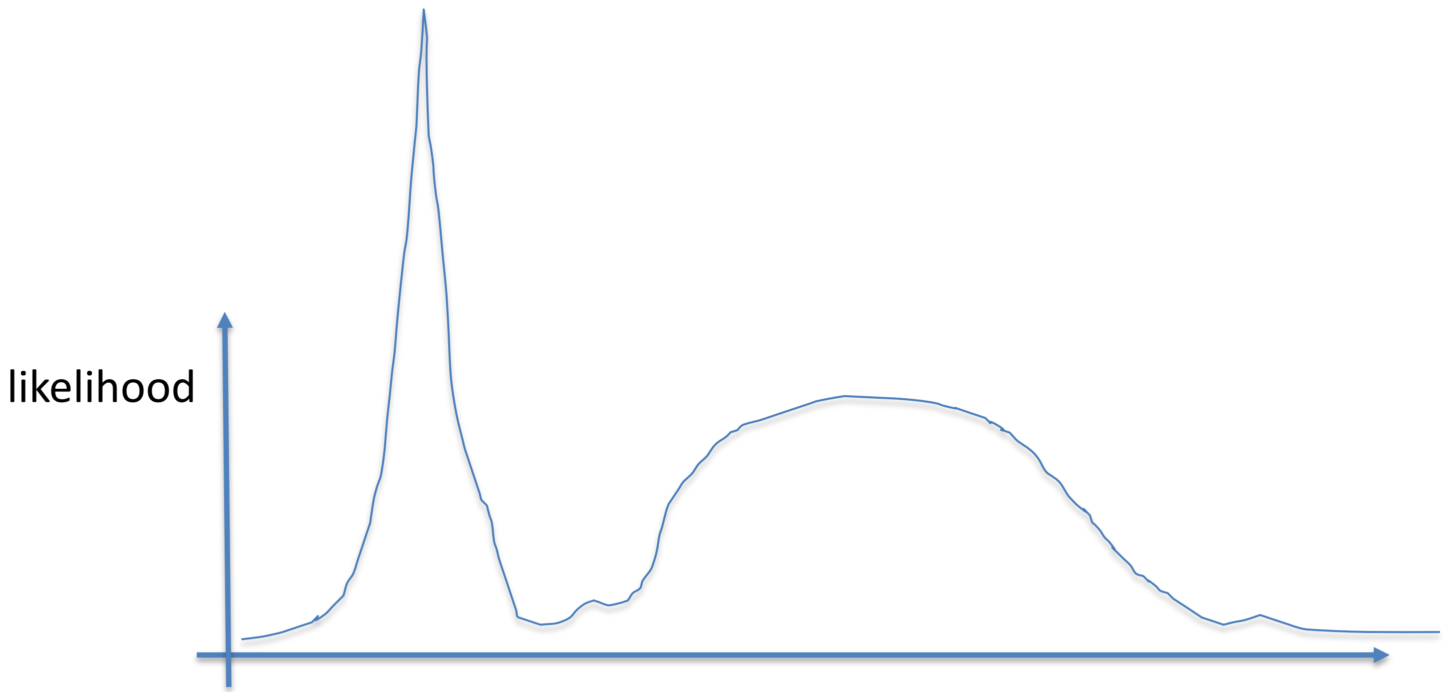


Figure 1: Comparison of reconstruction methods for in-distribution, sampling-shift, and anatomy-shift images. Our generative prior was trained *exclusively* on T2-weighted *brain* scans from the NYU fastMRI dataset, and the training of the generative prior was independent of the measurement model. Observers familiar with reading MR images [88] will notice that our reconstruction is similar in diagnostic value to E2E-Varnet, a supervised state of the art, when in-distribution and outperforms all competing methods out of distribution. In the latter case, competing methods often introduce artifacts that render them diagnostically unusable.

ML reconstruction has a problem for underrepresented classes



Our new results show that conditional sampling (Langevin Dynamics)
is the right way to generate, as opposed to ERM/ML reconstructions
Sample from P(x | y) as opposed to max P(x|y).
(email me for pointers)

Takeaways: Modeling high-dimensional distributions

- Use guided unsupervised methods **to create synthetic data that agrees with observations.**
- Search in the latent space to match the measurements
- Expand the range of generators as needed, depending on the number of measurements.
- DCGAN and older generators were very sensitive to cropping, color range, etc. We think we solve these problems with ILO and score-based models.
- Special care is needed on extrapolating bias in the training data or measurement errors/miscalibrations.

Theory: Expanding the sample complexity bounds from compressed sensing beyond sparsity to generative models.
Optimization Guarantees for non-convex GAN-projection problems
Robustness, **Quantization**, Different measurements, etc.

Conclusions

- Generative models are **powerful data-driven priors**
- Very **modular**, plug and play other boxes and back-prop through everything

- Open research directions:
- 1. Solving inverse problems with generative models– proofs for the optimization problems.
- 2. Imposing physical constraints on the generated data
- 3. Robustness to errors/corruptions in measurements.
  https://arxiv.org/abs/2006.09461
  Robust compressed sensing of generative models, A. Jalal et al. Outlier Detection using Generative Models with Theoretical Performance Guarantees, by Xu et al.
- 4. Fairness in inverse problems-– new interesting problems
- 5. MRI and other exciting medial imaging applications

- Papers, code and pre-trained models:
- http://users.ece.utexas.edu/~dimakis
- Twitter: @AlexGDimakis

References

CSGM: Compressed sensing using Generative models
A. Bora A. Jalal, E. Price, AGD, ICML 2017

ILO: Intermediate Layer Optimization for Inverse Problems using Deep Generative Models
ICML 2021.
Joseph Dean, Ajil Jalal, Giannis Daras, AGD.

MRI: Robust Compressed Sensing MRI with Deep Generative Priors
Ajil Jalal, Marius Arvinte, Giannis Daras, Eric Price, AGD, Jonathan I. Tamir
NeurIPS 2021
https://arxiv.org/abs/2108.01368

Fairness: Fairness for Image Generation with Uncertain Sensitive Attributes
Ajil Jalal, Sushrut Karmalkar, Jessica Hoffmann, AGD, Eric Price
ICML 2021
https://ajiljalal.github.io/fairness.html

survey:
Deep Learning Techniques for Inverse Problems in Imaging
Gregory Ongie, Ajil Jalal, Christopher A Metzler Richard G Baraniuk, Alexandros G Dimakis, Rebecca Willett,
Journal on Selected Areas in Information Theory (JSAIT), 2020.

- Fin

Pointers

Deep Generative models for inverse problems:

Compressed sensing using Generative models (Bora et al. ICML 2017)
1. How to solve inverse problems using a deep generative model G(z)
2. CSGM Guarantees through Set Restricted Eigenvalue conditions.

A central algorithmic challenge: Inverting deep generative models.

1.P. Hand and V. Voroninski, Global guarantees for enforcing deep priors by empirical risk, (COLT 2018)

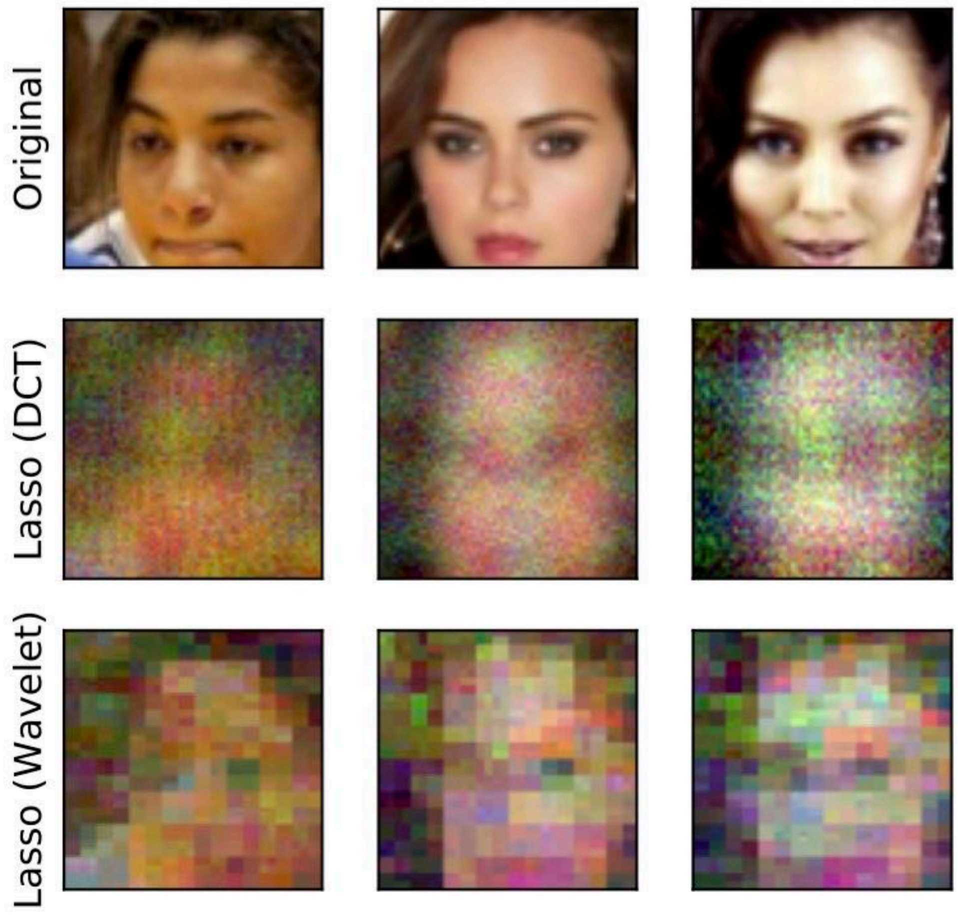
2.Inverting Deep Generative models, One layer at a time
Q. Lei, A. Jalal, I. Dhillon, A.G.D, NeurIPS 2019

3.Constant-Expansion Suffices for Compressed Sensing with Generative Priors.
C. Daskalakis, D. Rohatgi, M. Zampetakis

IA few results on **invertible** generative models.

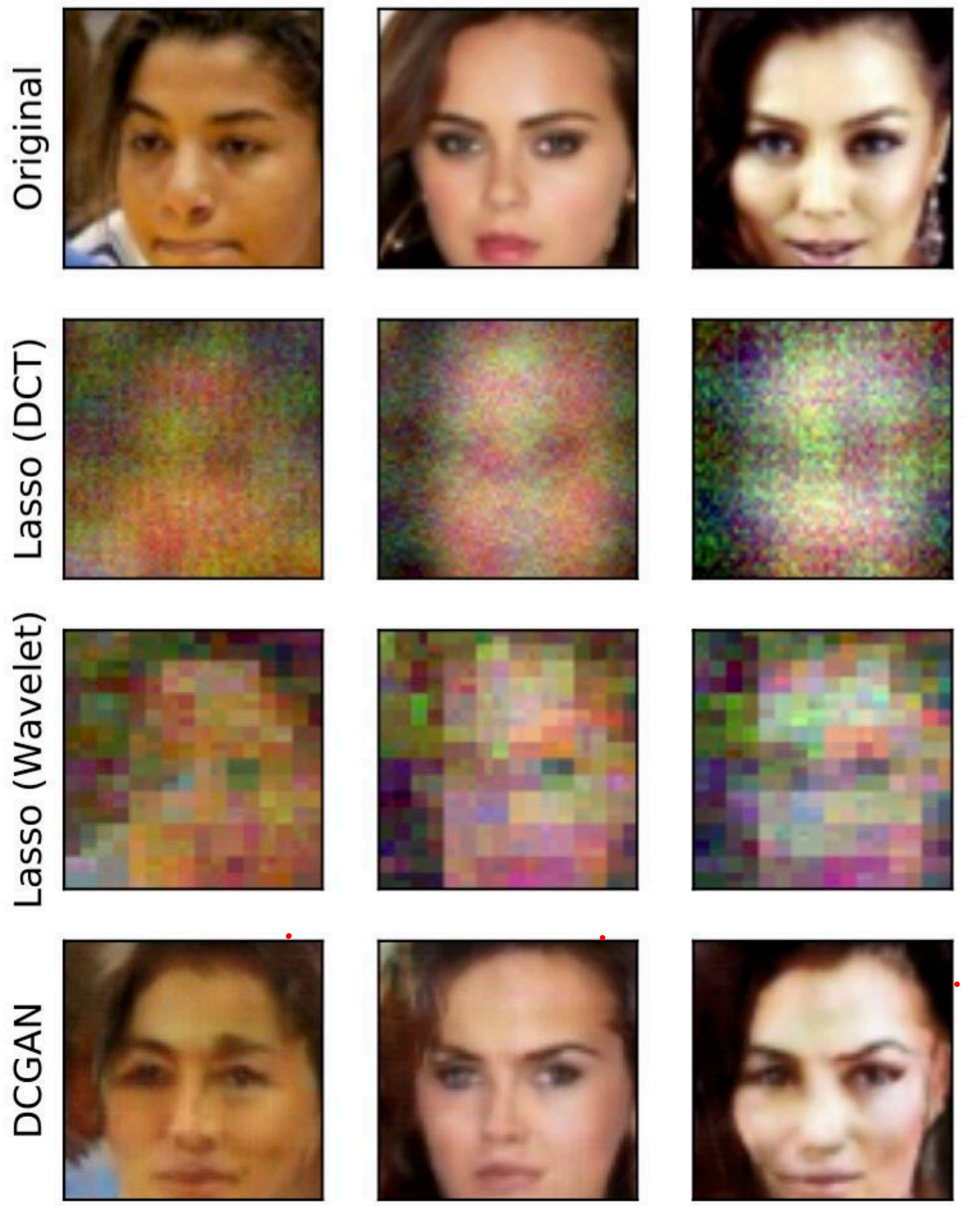
Conditional Sampling from Invertible Generative Models with Applications to Inverse Problems
by Erik Lindgren et al. https://arxiv.org/abs/2002.11743

Comparison to Lasso



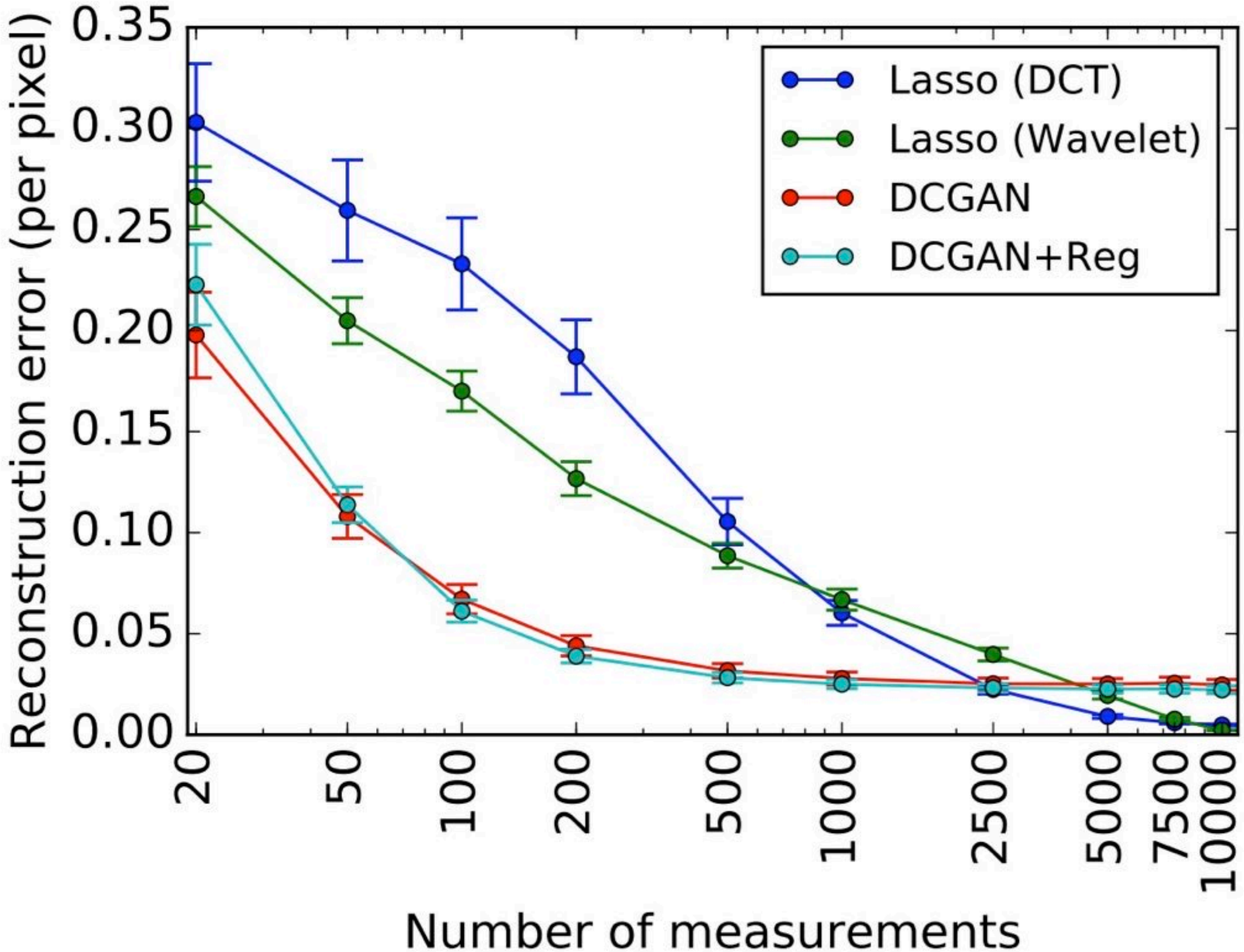
- m=500 random Gaussian measurements.
- n= 13k dimensional vectors.

Comparison to Lasso



- m=500 random Gaussian measurements.
- n= 13k dimensional vectors.

Comparison to Lasso



(b) Results on celebA

Compressed sensing using generative models, Bora et al. ICML 2017.

Related work

- Significant prior work on structure beyond sparsity
- **Model-based CS** (Baraniuk et al., Cevher et al., Hegde et al., Gilbert et al. , Duarte & Eldar)
- **Projections on Manifolds:**
- Baraniuk & Wakin (2009) Random projections of smooth manifolds. Eftekhari & Wakin (2015)
- **Deep network models:**
- Mousavi, Dasarathy, Baraniuk
- Chang, J., Li, C., Poczos, B., Kumar, B., and Sankaranarayanan, ICCV 2017

Theory results

Talk Outline

Deep Generative models for inverse problems:

~~Compressed sensing using Generative models (Bora et al. ICML 2017)~~
~~1. How to solve inverse problems using a deep generative model G(z)~~
2. CSGM Guarantees through Set Restricted Eigenvalue conditions.

Theory for Inverting deep generative models.

1.P. Hand and V. Voroninski, Global guarantees for enforcing deep priors by empirical risk, (COLT 2018)
2.Inverting Deep Generative models, One layer at a time
Q. Lei, A. Jalal, I. Dhillon, A.G.D, NeurIPS 2019

New Algorithmic Developments:
Intermediate Layer Optimization + Perceptual Distances

Intermediate Layer Optimization for Inverse Problems using Deep Generative Models
Deep-Inverse Workshop, NeurIPS 2020. , Joseph Dean, Giannis Daras, AD.

If we have time: A few results on **invertible** generative models.

Conditional Sampling from Invertible Generative Models with Applications to Inverse Problems
by Erik Lindgren et al. https://arxiv.org/abs/2002.11743

Theory results

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$

Theory results

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$
- **Theorem 1:** If A is iid N(0, 1/m) with $m = O(kd \log n)$
- Then the reconstruction is close to optimal:

$$||G(\hat{z}) - x^*||_2 \leq C \min_z ||G(z) - x^*||$$

Theory results

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$
- **Theorem 1:** If A is iid N(0, 1/m) with $m = O(kd \log n)$
- Then the reconstruction is close to optimal:

$$||G(\hat{z}) - x^*||_2 \leq C \min_z ||G(z) - x^*||$$

- (Reconstruction accuracy proportional to model accuracy)
- **Thm2:** More general result: m = O(k log L) measurements for any L-Lipschitz function G(z)

Compressed sensing using generative models, Bora et al. ICML 2017.

- Let $y = Ax$
- Solve $\hat{z} =$
- **Theorem 1**
- Then the

$||G($

- (Reconstruction
- **Thm2:** More gen... ny
  L-Lipschitz functio

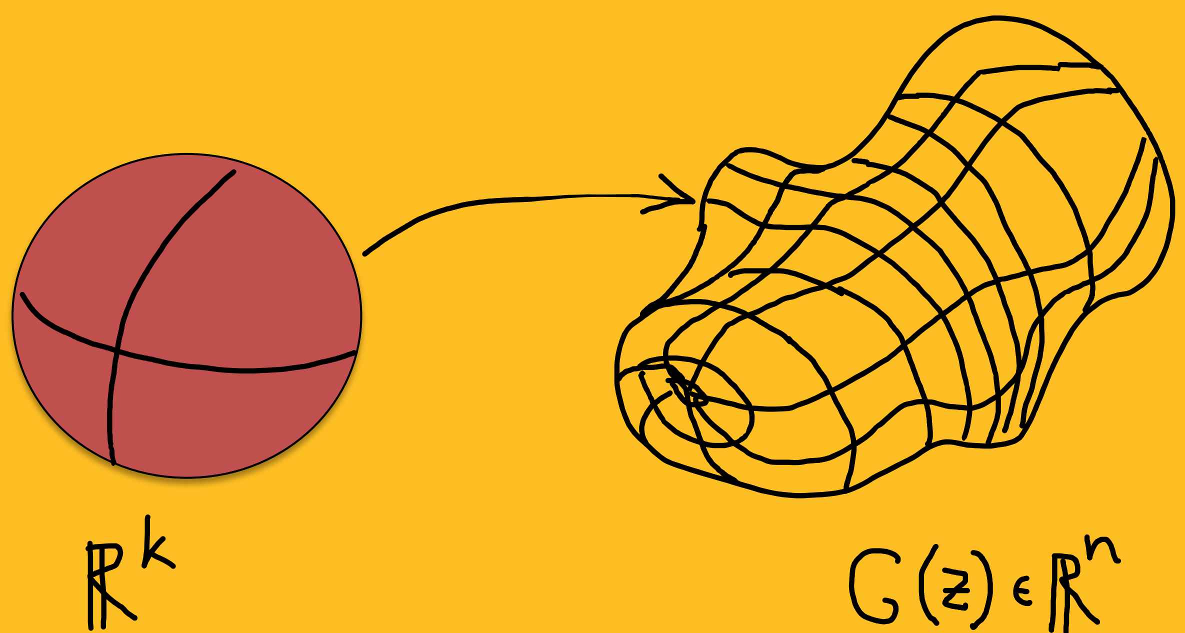
RIP property for a measurement matrix: all sparse vectors are far from the nullspace of measurement matrix

We define a **set restricted eigenvalue condition (S-REC)** that asks that **the differences of pairs of generated images is far from the nullspace.**

Key Lemma: Random matrices with m=k logL rows will satisfy S-REC whp.

- Let y
- Solve $\hat{z} =$
- Th
-

Key Lemma: Random matrices with m=k logL rows will satisfy S-REC whp.
How to bound metric entropy (aka log Covering number) of generator range

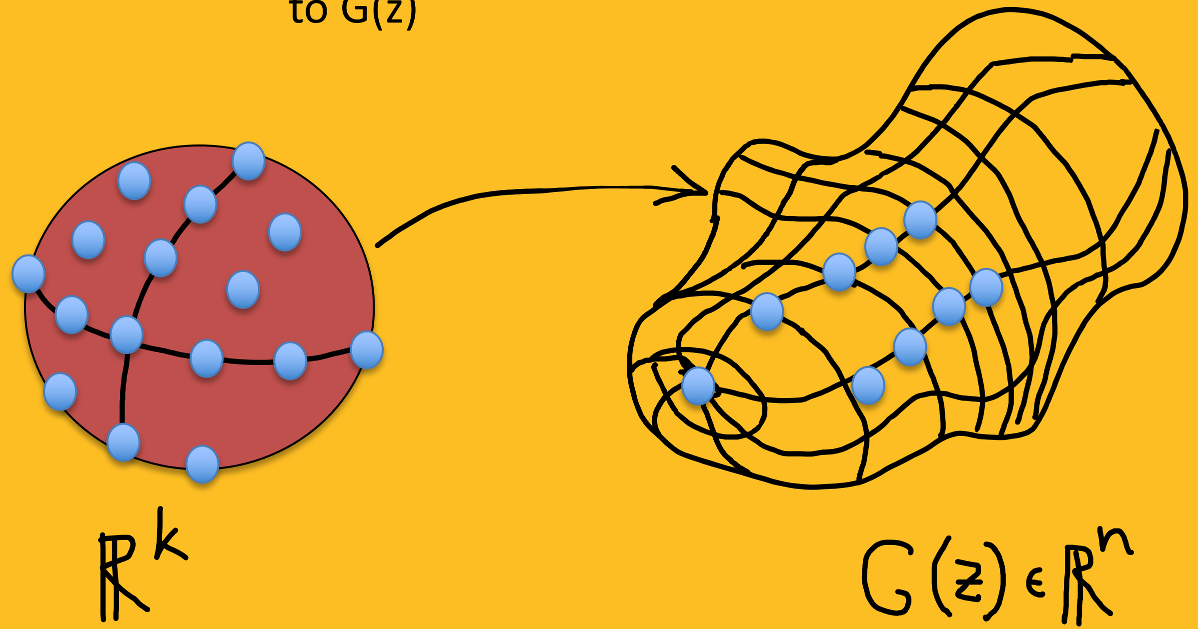


y

- Let y
- Solve $\hat{z} =$
- **Th**

Key Lemma: Random matrices with
m=k logL rows will
satisfy S-REC whp.
How to bound metric entropy (aka log Covering
number) of generator range

Distance distortion L due
to G(z)



Main results

Theorem 1.1. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be a generative model from a d-layer neural network using ReLU activations. Let $A \in \mathbb{R}^{m \times n}$ be a random Gaussian matrix for $m = O(kd \log n)$, scaled so $A_{i,j} \sim N(0, 1/m)$. For any $x^* \in \mathbb{R}^n$ and any observation $y = Ax^* + \eta$, let $\widehat{z}$ minimize $\|y - AG(z)\|_2$ to within additive ϵ of the optimum. Then with $1 - e^{-\Omega(m)}$ probability,

$$\|G(\widehat{z}) - x^*\|_2 \le 6 \min_{z^* \in \mathbb{R}^k} \|G(z^*) - x^*\|_2 + 3\|\eta\|_2 + 2\epsilon.$$

- The first and second term are essentially necessary.
- The third term is the extra penalty ε for gradient descent sub-optimality.

Main results

Theorem 1.1. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be a generative model from a d-layer neural network using ReLU activations. Let $A \in \mathbb{R}^{m \times n}$ be a random Gaussian matrix for $m = O(kd \log n)$, scaled so $A_{i,j} \sim N(0, 1/m)$. For any $x^* \in \mathbb{R}^n$ and any observation $y = Ax^* + \eta$, let $\widehat{z}$ minimize $\|y - AG(z)\|_2$ to within additive ϵ of the optimum. Then with $1 - e^{-\Omega(m)}$ probability,

$$\|G(\widehat{z}) - x^*\|_2 \leq 6 \min_{z^* \in \mathbb{R}^k} \|G(z^*) - x^*\|_2 + 3\|\eta\|_2 + 2\epsilon.$$

Representation error

noise

optimization error

- The first and second term are essentially necessary.
- The third term is the extra penalty ε for gradient descent sub-optimality.

Main results

Theorem 1.2. Let $G : \mathbb{R}^k \to \mathbb{R}^n$ be an L-Lipschitz function. Let $A \in \mathbb{R}^{m \times n}$ be a random Gaussian matrix for $m = O(k \log \frac{Lr}{\delta})$, scaled so $A_{i,j} \sim N(0, 1/m)$. For any $x^* \in \mathbb{R}^n$ and any observation $y = Ax^* + \eta$, let $\widehat{z}$ minimize $\|y - AG(z)\|_2$ to within additive ϵ of the optimum over vectors with $\|\widehat{z}\|_2 \leq r$. Then with $1 - e^{-\Omega(m)}$ probability,

$$\|G(\widehat{z}) - x^*\|_2 \leq 6 \min_{\substack{z^* \in \mathbb{R}^k \\ \|z^*\|_2 \leq r}} \|G(z^*) - x^*\|_2 + 3\|\eta\|_2 + 2\epsilon + 2\delta.$$

- For general L-Lipschitz functions.
- Minimize only over z vectors within a ball.
- Assuming poly(n) bounded weights: $L = n^{O(d)}$, $\delta = 1/n^{O(d)}$

Proof technology

Architecture of compressed sensing proofs for Lasso:

Lemma 1: A random Gaussian measurement matrix has **RIP/REC** whp

Lemma 2: Lasso works for matrices that have **RIP/REC**.
Lasso recovers a x_{hat} close to x^*

Proof technology

For a generative model defining a subset of images S:

Lemma 1: A random Gaussian measurement matrix has **S-REC** whp for sufficient measurements.

Lemma 2: The optimum of the squared loss minimization recovers a z_{hat} close to z^* **if A has S-REC.**

Proof technology

Why is the Restricted Eigenvalue Condition (REC) needed?

Lasso solves:

$$\min_{s.t.:||Ax-y||_2<\epsilon} ||x||_1$$

If there is a sparse vector x in the nullspace of A then this fails.

Proof technology

Why is the Restricted Eigenvalue Condition (REC) needed?

Lasso solves:

$$\min_{s.t.:||Ax-y||_2<\epsilon} ||x||_1$$

If there is a sparse vector x in the nullspace of A then this fails.

REC: All approximately k-sparse vectors x are far from the nullspace:

$$\gamma||x||_2 \leq ||Ax|||_2$$

A vector x is approximately k-sparse if there exists a set of k coordinates S such that

$$||x_S||_1 \geq ||x_{S^c}||_1$$

Proof technology

Unfortunate coincidence: The difference of two k-sparse vectors is 2k sparse.

But the difference of two natural images is not natural.

The correct way to state REC (That generalizes to our S-REC) is

For **any two k-sparse** vectors x1,x2 , their difference is far from the nullspace:

$$\gamma||x_1 - x_2||_2 \leq ||A(x_1 - x_2)||_2$$

Proof technology

Our Set-Restricted Eigenvalue Condition (**S-REC**). For any set
$S \subset \mathbb{R}^n$

A matrix A satisfies **S-REC** if for all x_1, x_2 in S

For **any two natural images,** their difference is far from the nullspace of A:

$$\gamma ||x_1 - x_2||_2 \leq ||A(x_1 - x_2)||_2$$

Proof technology

Our Set-Restricted Eigenvalue Condition (**S-REC**). For any set

$S \subset \mathbb{R}^n$

A matrix A satisfies S-REC if for all x_1, x_2 in S

The difference of two natural images is far from the nullspace of A:

$$\gamma||x_1 - x_2||_2 \leq ||A(x_1 - x_2)||_2$$

- Lemma1: If the set S is the range of a generative model then m= O (k logL) measurements suffice to make a gaussian iid matrix S-REC whp.
- Lemma2: If the matrix has S-REC then squared loss optimizer z_{hat} must be close to z^*

Talk Outline

Deep Generative models for inverse problems:

~~Compressed sensing using Generative models (Bora et al. ICML 2017)~~
~~1. How to solve inverse problems using a deep generative model G(z)~~
~~2. CSGM Guarantees through Set Restricted Eigenvalue conditions.~~

Theory for Inverting deep generative models.
1.P. Hand and V. Voroninski, Global guarantees for enforcing deep priors by empirical risk, (COLT 2018)
2.Inverting Deep Generative models, One layer at a time
Q. Lei, A. Jalal, I. Dhillon, A.G.D, NeurIPS 2019

New Algorithmic Developments:
Intermediate Layer Optimization + Perceptual Distances
Intermediate Layer Optimization for Inverse Problems using Deep Generative Models
Deep-Inverse Workshop, NeurIPS 2020. , Joseph Dean, Giannis Daras, AD.

If we have time: A few results on **invertible** generative models.
Conditional Sampling from Invertible Generative Models with Applications to Inverse Problems
by Erik Lindgren et al. https://arxiv.org/abs/2002.11743

Theory for Optimization

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$ Open: How to do efficiently ? (under the right conditions)
- **Theorem 1:** If A is iid N(0, 1/m) with $m = O(kd \log n)$
- Then the reconstruction is close to optimal:

$$||G(\hat{z}) - x^*||_2 \leq C \min_z ||G(z) - x^*||$$

- (Reconstruction accuracy proportional to model accuracy)
- **Thm2:** More general result: m = O(k log L) measurements for any L-Lipschitz function G(z)

Compressed sensing using generative models, Bora et al. ICML 2017.

Theory for Optimization

- Let $y = Ax^* + \eta$
- Solve $\hat{z} = \min_z ||y - AG(z)||$

Open: How to do efficiently ?
(under the right conditions)

For generators with random iid weights, gradient descent provably solves this problem!
(Assuming each layer is logk factor bigger compared to previous one).

Hand and Voroninski
Global guarantees for enforcing deep priors by empirical risk (COLT 2018)

Leong, Hand, Voroninski
Phase Retrieval Under a Generative Prior (NeurIPS 2018)