Applications and Research Frontiers in Deep Generative Models

CS 236: Deep Generative Models

Stanford University

Presentation Lineup

- **Buffered Stochastic Variational Inference.** *Rui Shu*
- **Neural Joint Source-Channel Coding.** *Kristy Choi*
- **Learning Fair and Controllable Representations.** *Jiaming Song*
- **Constructing Unrestricted Adversarial Examples with Generative Models.** *Yang Song*
- **CycleGAN, a Master of Steganography.** *Casey Chu*

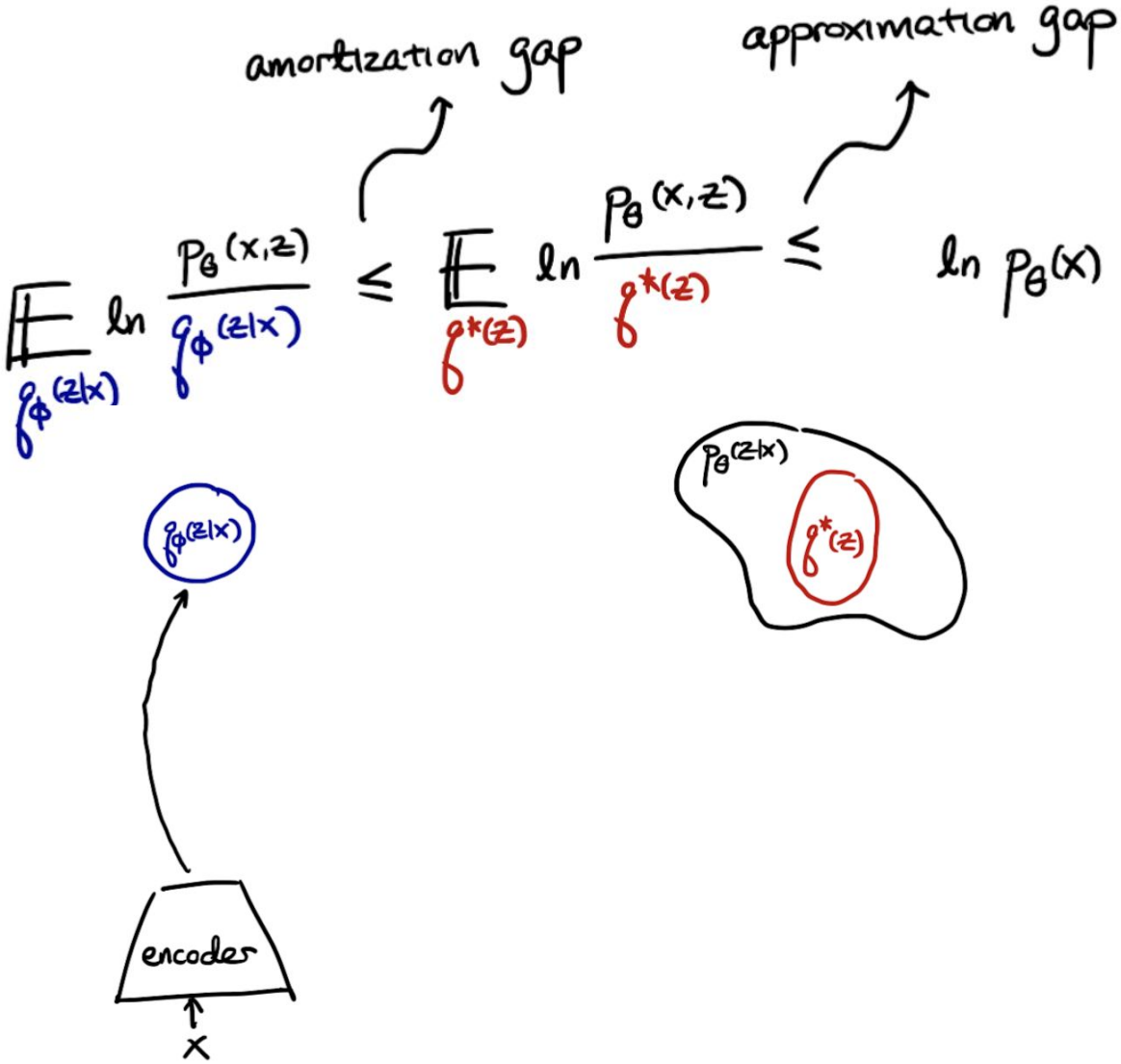
Buffered Stochastic Variational Inference

Rui Shu

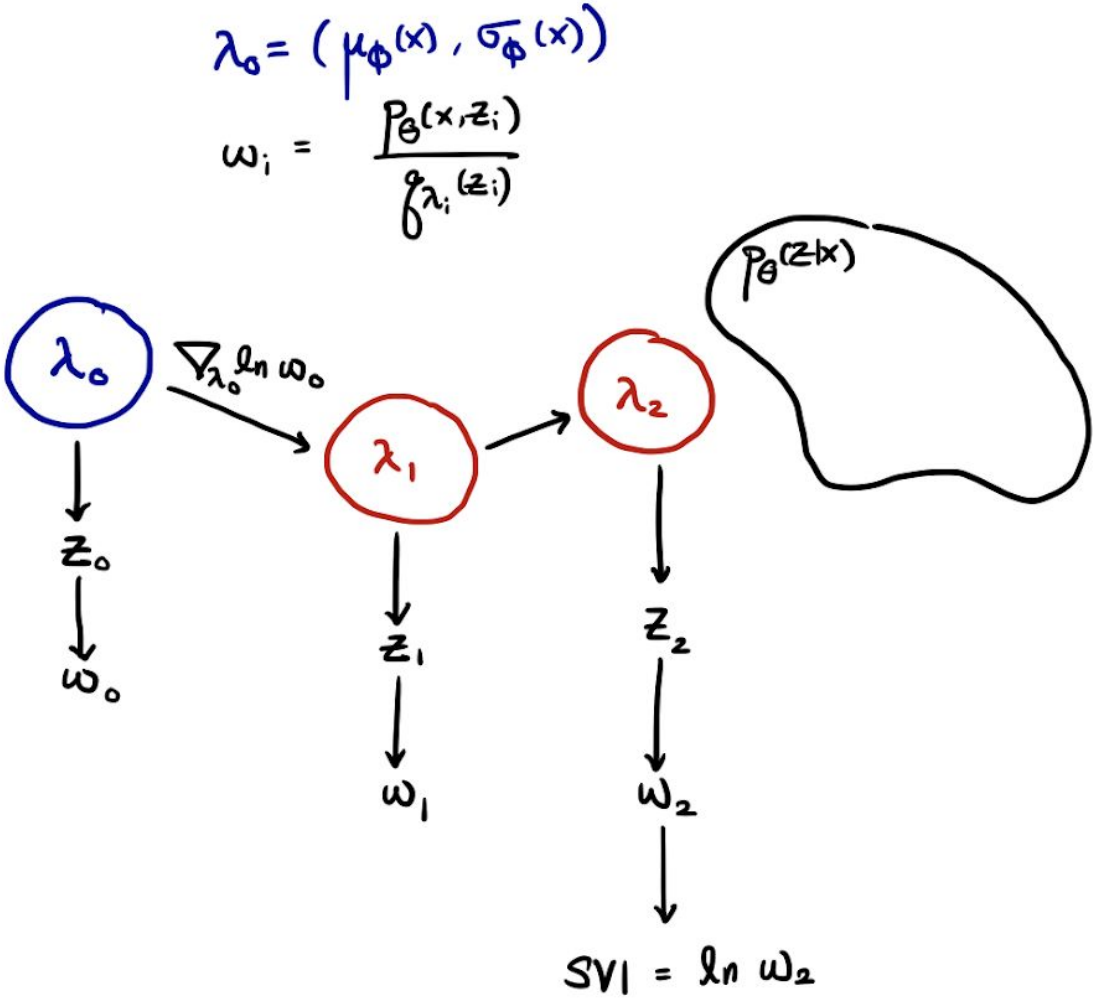
Hung Bui, Jay Whang, Stefano Ermon

Stanford University

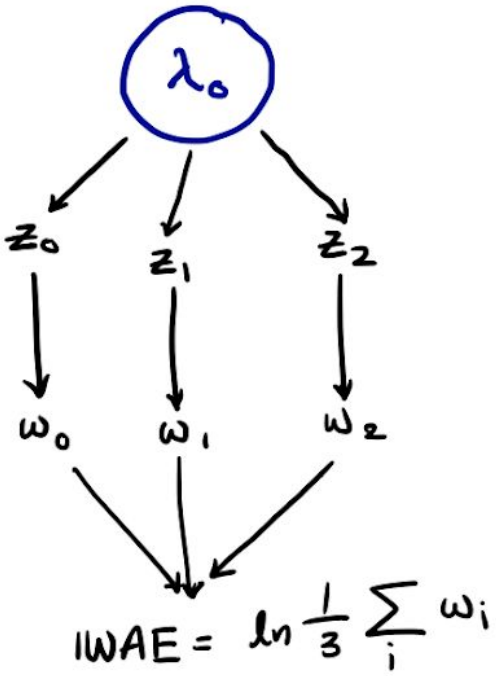
Amortization Gap

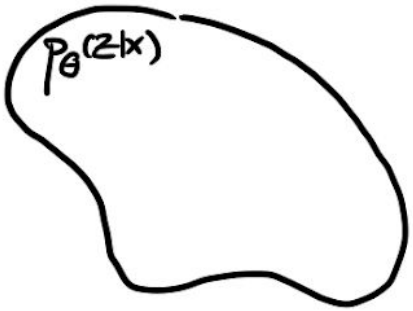


Stochastic Variational Inference

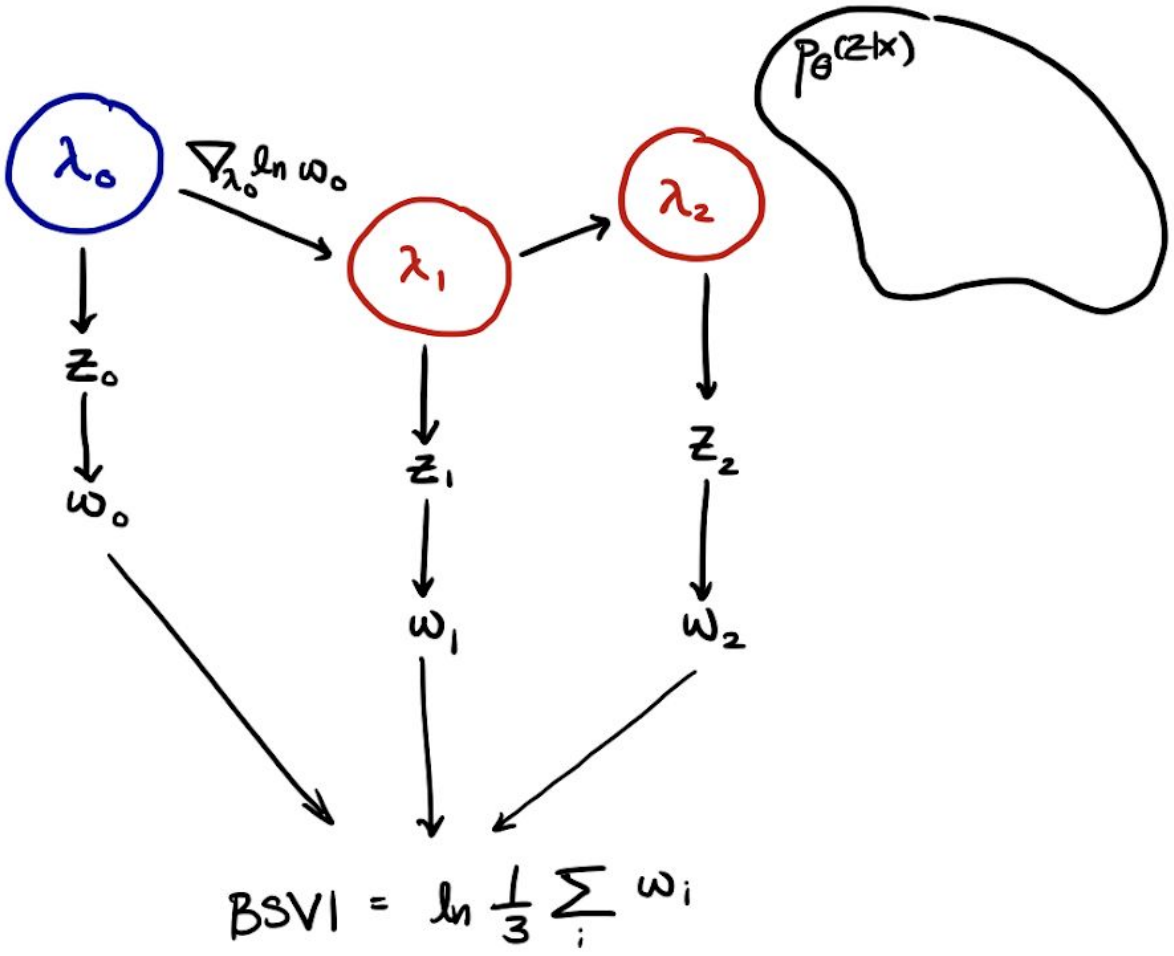


Importance-Weighted Autoencoder

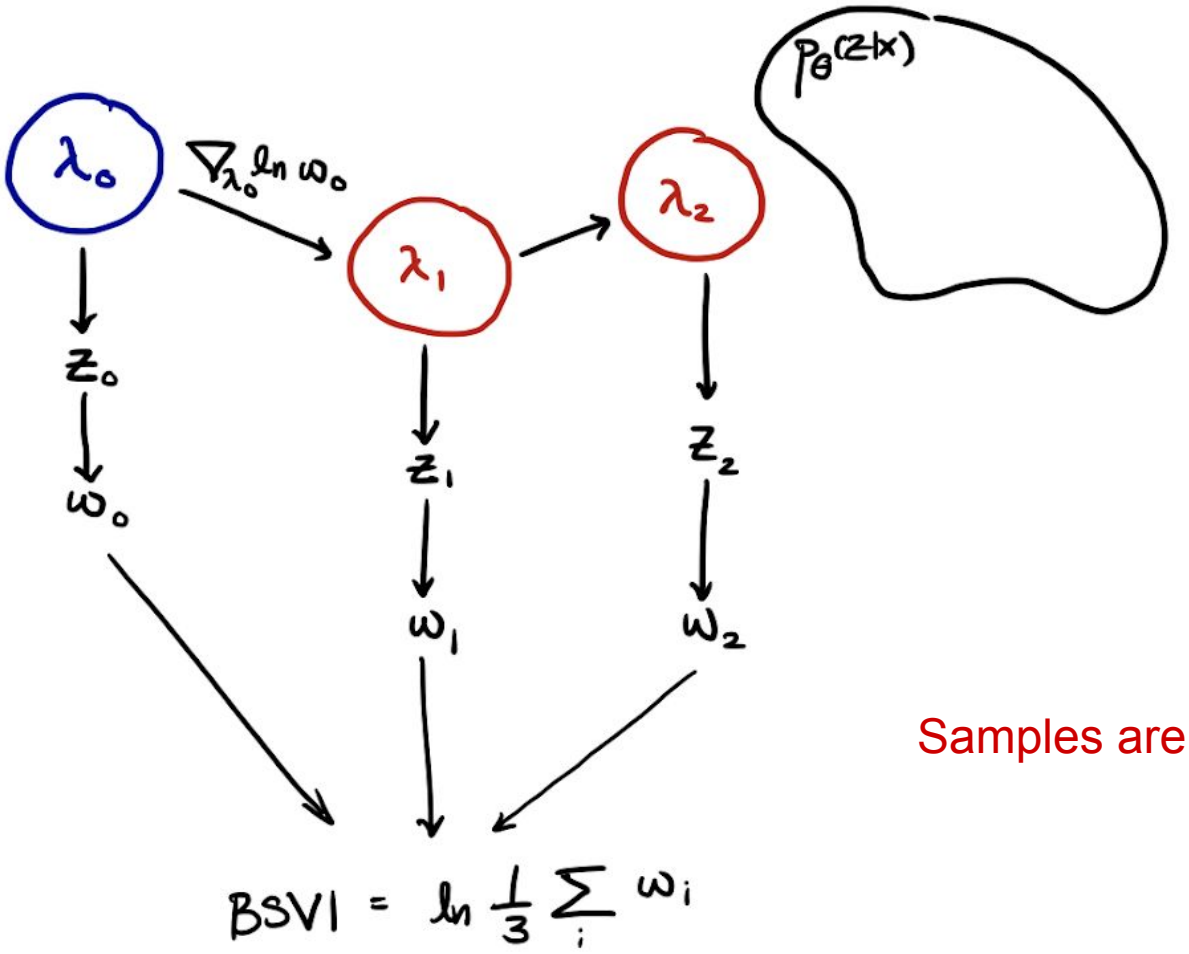




Buffered Stochastic Variational Inference

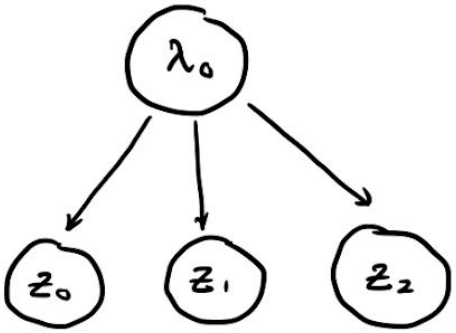


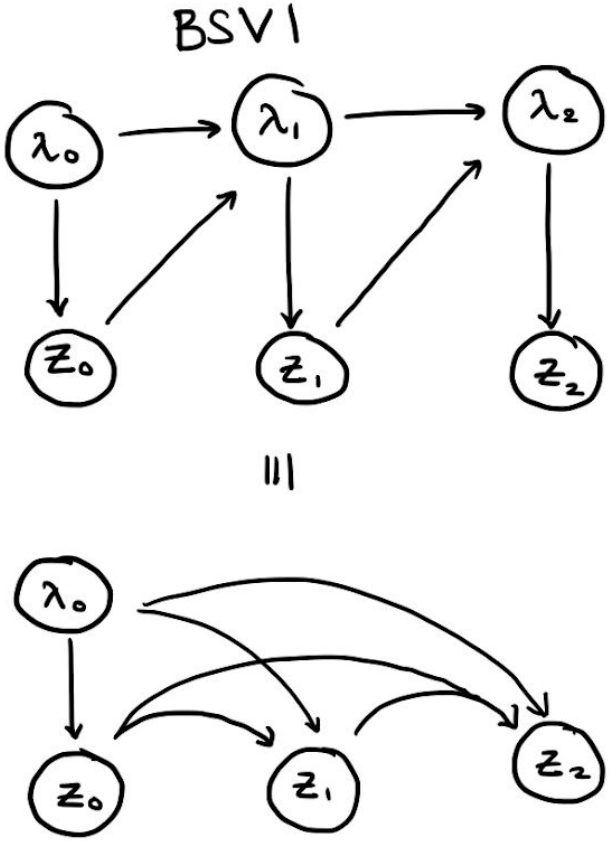
Is BSVI a Valid Lowerbound?



Samples are **not i.i.d.**

Is BSVI a Valid Lowerbound?

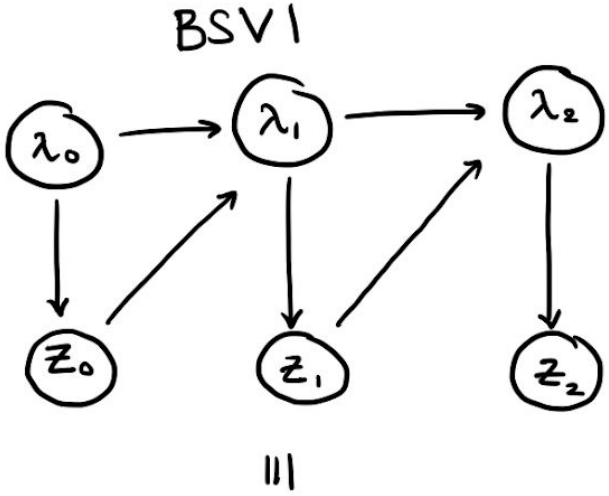




$$q(z_{1:K}|x) = \prod_i q(z_i|x)$$

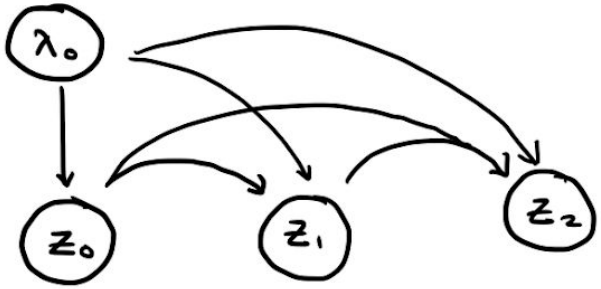
$$q(z_{1:K}|x) = \prod_i q(z_i|z_{<i}, x)$$

Is BSVI a Valid Lowerbound?



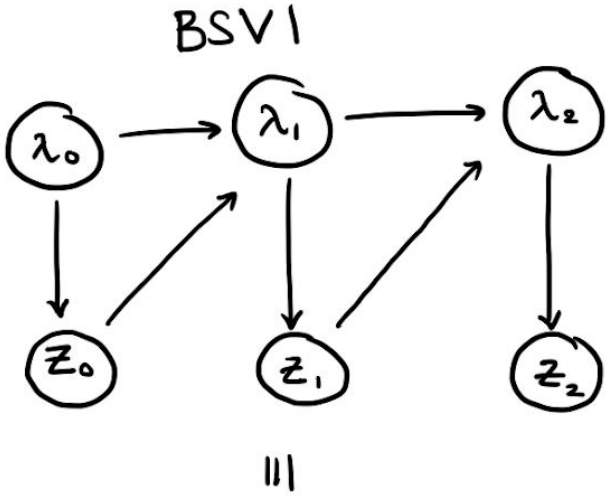
$$\ln p_\theta(x) = \ln \mathop{\mathbb{E}}_{q(z_{1:k})} \frac{1}{k} \sum_i \frac{p_\theta(x, z_i)}{q(z_i \mid z_{<i})}$$

$$\geq \mathop{\mathbb{E}}_{q(z_{1:k})} \ln \frac{1}{k} \sum_i \frac{p_\theta(x, z_i)}{q(z_i \mid z_{<i})} = \text{BSVI}$$



$$q(z_{1:k} \mid x) = \prod_i q(z_i \mid z_{<i}, x)$$

Is BSVI a Valid Lowerbound?

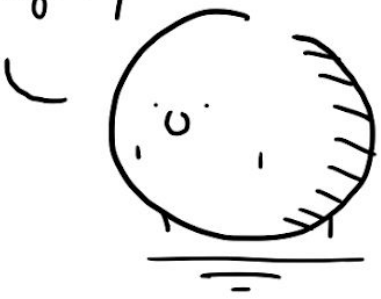


$$q(z_{1:k}|x) = \prod_i q(z_i|z_{<i}, x)$$

$$\ln p_\theta(x) = \ln \mathop{\mathbb{E}}_{q(z_{1:k})} \frac{1}{k}\sum_i \frac{p_\theta(x, z_i)}{q(z_i|z_{<i})}$$

$$\geq \mathop{\mathbb{E}}_{q(z_{1:k})} \ln \frac{1}{k}\sum_i \frac{p_\theta(x, z_i)}{q(z_i|z_{<i})} = \text{BSVI}$$

everything is just Jensen's Inequality



BSVI versus SVI

Inputs: $\mathcal{D} = \{x^{(1)}, \ldots, x^{(n)}\}$.
for $t = 1 \ldots T$ **do**
  $x \sim \mathcal{D}$
  $\lambda_0 \leftarrow f_{\phi_t}(x)$
  for $i = 0 \ldots k$ **do**
    $z_i \sim q(z\,;\lambda_i)$ ▷ reparameterize as $z_{\lambda_i}(\epsilon)$
    $w(z\,;\lambda_i, \theta) \leftarrow p_\theta(x, z_i)/q(z_i\,;\lambda_i)$
    if $i < k$ **then**
      $\lambda_{i+1} \leftarrow \lceil \lambda_i + \eta \nabla_{\lambda_i} \ln w(z\,;\lambda_i, \theta) \rceil$
    end if
  end for
  $\phi_{t+1} \leftarrow \phi_t + \nabla_{\phi_t} \ln w(z_0\,;\lambda_0, \theta_t)$
  if Train with SVI **then**
    $\theta_{t+1} \leftarrow \theta_t + \nabla_{\theta_t} \ln w(z_k\,;\lambda_k, \theta_t)$
  else if Train with BSVI **then**
    $\theta_{t+1} \leftarrow \theta_t + \nabla_{\theta_t} \ln \sum_i \pi_i w(z_i\,;\lambda_i, \theta_t)$
  end if
end for

BSVI versus SVI

Inputs: $\mathcal{D} = \{x^{(1)}, \ldots, x^{(n)}\}$.
for $t = 1 \ldots T$ **do**
  $x \sim \mathcal{D}$
  $\lambda_0 \leftarrow f_{\phi_t}(x)$
  for $i = 0 \ldots k$ **do**
    $z_i \sim q(z\,;\lambda_i)$ ▷ reparameterize as $z_{\lambda_i}(\epsilon)$
    $w(z\,;\lambda_i,\theta) \leftarrow p_\theta(x, z_i)/q(z_i\,;\lambda_i)$
    if $i < k$ **then**
      $\lambda_{i+1} \leftarrow \lceil \lambda_i + \eta \nabla_{\lambda_i} \ln w(z\,;\lambda_i,\theta) \rceil$
    end if
  end for
  $\phi_{t+1} \leftarrow \phi_t + \nabla_{\phi_t} \ln w(z_0\,;\lambda_0,\theta_t)$
  if Train with SVI **then**
    $\theta_{t+1} \leftarrow \theta_t + \nabla_{\theta_t} \ln w(z_k\,;\lambda_k,\theta_t)$
  else if Train with BSVI **then**
    $\theta_{t+1} \leftarrow \theta_t + \nabla_{\theta_t} \ln \sum_i \pi_i w(z_i\,;\lambda_i,\theta_t)$
  end if
end for

Monte Carlo approx. of gradient makes BSVI **as fast as** SVI

Performance

Table 1: Test set performance on the Omniglot dataset. Note that $k = 10$.

Model	**Log-likelihood**
VAE	-89.83 $\pm$ 0.03
IWAE-k	-89.02 $\pm$ 0.05
SVI-k	-89.65 $\pm$ 0.06
BSVI-k	**-88.80 $\pm$ 0.03**

Table 2: Test set performance on the grayscale SVHN dataset.

Model	**Log-likelihood**
VAE	-2203 $\pm$ 14.95
IWAE-k	-2149 $\pm$ 10.11
SVI-k	-2074 $\pm$ 10.46
BSVI-k	**-2060 $\pm$ 3.54**

Table 3: Test set performance on the FashionMNIST dataset.

Model	**Log-likelihood**
VAE	-1734.34 $\pm$ 0.71
IWAE-k	-1705.90 $\pm$ 0.29
SVI-k	-1708.31 $\pm$ 2.32
BSVI-k	**-1699.86 $\pm$ 0.22**

Conclusion

If worried about amortization gap: use SVI

If using SVI: use BSVI instead

Use BSVI instead of IWAE for during evaluation

Neural Joint Source-Channel Coding

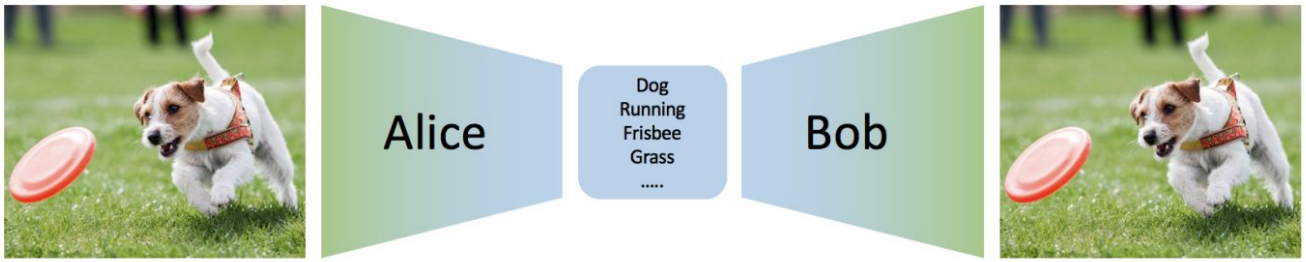
Kristy Choi

w/ Kedar Tatwawadi, Tsachy Weissman, Stefano Ermon

Stanford University

The Communication Game

- Alice wants to communicate an image to Bob
  - But she can only do so over a noisy channel
- Shannon (1948) proved that in this scenario, the optimal thing to do is to:
  - **source code:** compress the image into a bitstring
  - **channel code:** add redundancy to the bitstring
  - *independently of each other*



Joint Source-Channel Coding (JSCC)

- Why is this problem hard?
- **Infinitely long messages:** can use as many bits as you want to compress/channel code
- **Finite # bits:** need to worry about bit allocation
  - Balance distortion from 2 sources
- AND need to learn the function that does the (1) compression and (2) channel coding for you

Problem Setup

- $\mathcal{X}$: space of possible inputs
- $p_{\text{data}}(x)$: source distribution over $x \in \mathcal{X}$
- Encode $x \sim p_{\text{data}}(x)$ into $\hat{y} = \{0, 1\}^m$
- Noisy channel corrupts codes $\hat{y}$ to become a noisy code y
- **Goal:** minimize overall distortion in L1 or L2 norm: $||x - \hat{x}||$ while keeping the message length *m* small

Learning Objective

- Mutual information maximization
- $\max_{\phi} I(X, Y; \phi, \epsilon)$
  - Want codewords Y to capture as much information about the data X as possible, *even after they've been corrupted!*
  - Also the **capacity** of the channel
- Lower bound turns into something nice:

$$\max_{\phi} I(X, Y; \phi, \epsilon) = H(X) - H(X|Y; \phi, \epsilon)$$

$$\geq \mathbb{E}_{x \sim p_{\text{data}}(x)} \mathbb{E}_{y \sim q_{\text{noisy\_enc}}(y|x;\epsilon,\phi)} [\log p_{\text{decoder}}(x|y;\theta)] + \text{const.}$$

Reconstruction loss!

Coding Process

$$X \rightarrow \hat{Y} \rightarrow Y \rightarrow \hat{X}$$

- We can model everything as a graphical model!

$$p(x, \hat{y}, y, \hat{x}) = p_{\text{data}}(x) q_{\text{encoder}}(\hat{y}|x;\phi) p_{\text{channel}}(y|\hat{y};\epsilon) p_{\text{decoder}}(\hat{x}|y;\theta)$$

- Channel model $p_{\text{channel}}(y|\hat{y};\epsilon)$
- Encoder $q_{\text{encoder}}(\hat{y}|x;\phi)$
- Noisy encoder (after corruption) $q_{\text{noisy\_enc}}(y|x;\epsilon,\phi)$
  - Trick is to marginalize out the codes $\hat{y}$

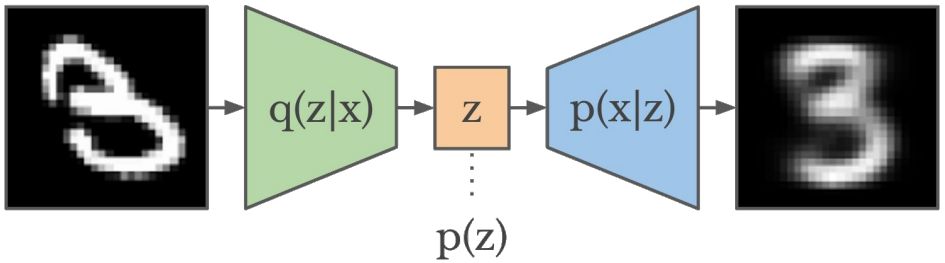


Figure from Danijar Hafner

Optimization Procedure

- Our latent variables *y* are discrete → hard!
- Use VIMCO*
  - Essentially the *K*-sample analogue to NVIL
  - Draw multiple samples from your inference network, tighter lower bound

$$\mathcal{L}^K(\phi, \theta; x, \epsilon) = \max_{\theta,\phi} \sum_{x \in \mathcal{D}} \mathbb{E}_{y^{1:K} \sim q_{\text{noisy\_enc}}(y|x,\epsilon,\phi)} \left[\log \frac{1}{K} \sum_{i=1}^{K} p_{\text{decoder}}(x|y^i;\theta) \right]$$

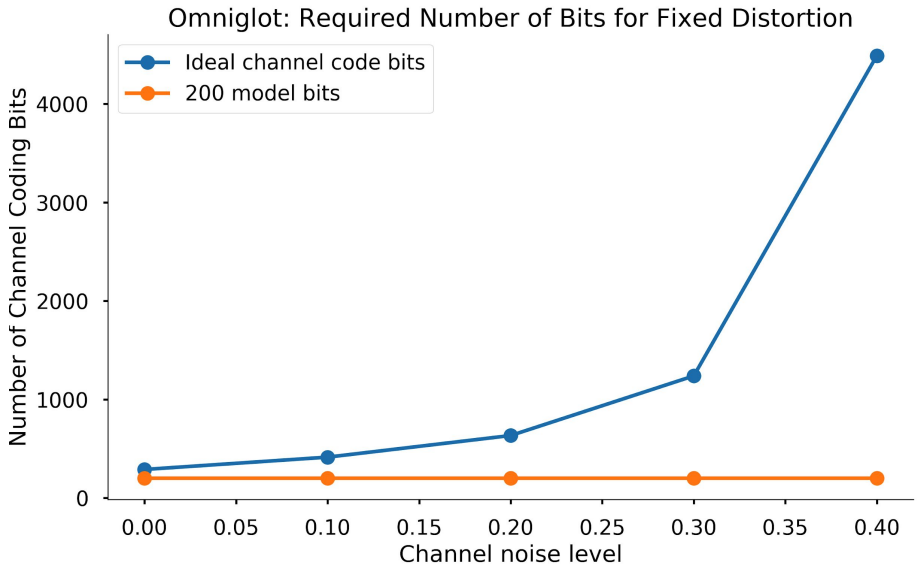
Multiple draws from *y*

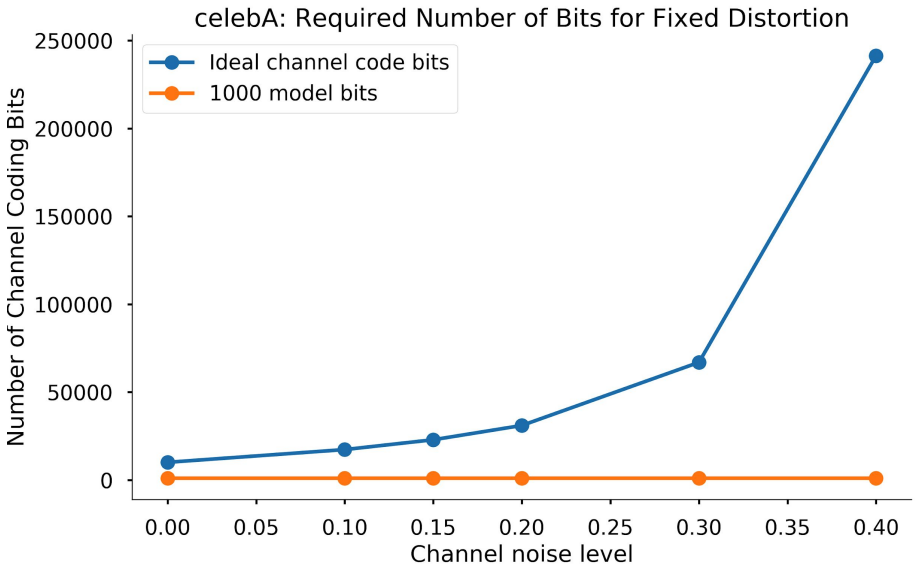
Multiple reconstruction loss terms

*Mnih and Rezende, Variational Inference for Monte Carlo Objectives.ICML 2016

Comparison against Ideal Codes

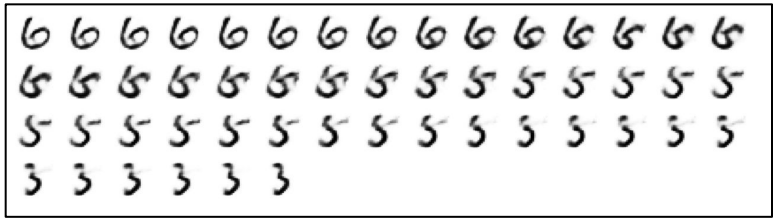
- We need *a lot less bits* to get the same level of distortion, even when we pair JPEG with an ideal channel code

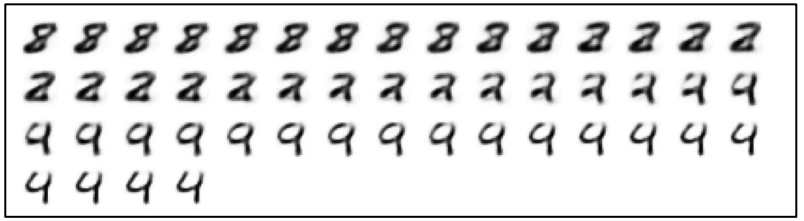




Robust Representation Learning

- Encoded redundancies: interpolation in latent space by bit-flip



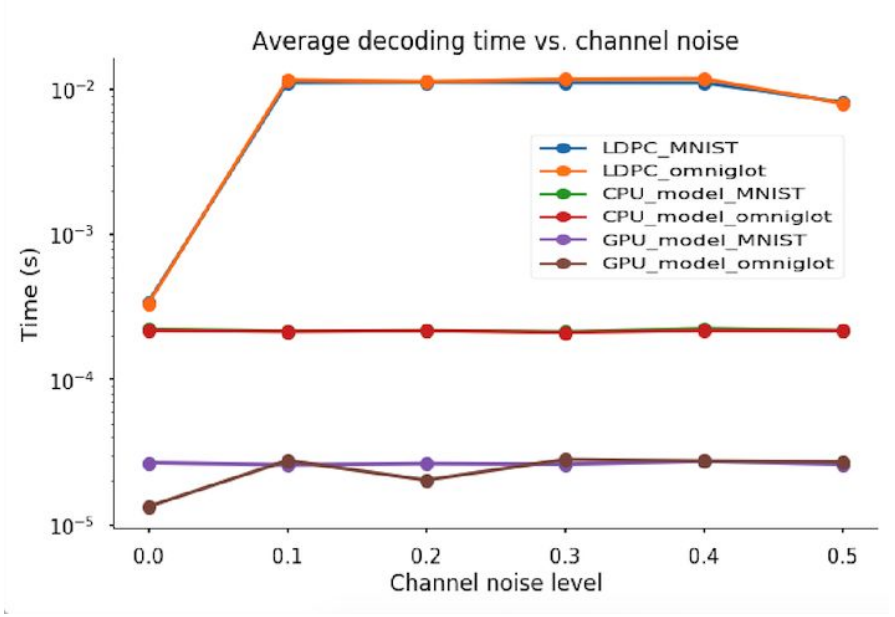


- Improved downstream classification with noisy MNIST

Noise ϵ	KNN	DT	RF	MLP	AdaB	NB	QDA	SVM
0	**0.95/0.86**	0.65/0.54	0.71/0.59	0.93/0.87	0.72/0.65	0.75/0.65	0.56/0.28	0.88/0.81
0.1	0.95/0.86	0.65/0.59	0.74/0.65	**0.934**/0.88	0.74/0.72	0.83/0.77	**0.94/0.90**	0.92/0.84
0.2	0.94/0.86	**0.78/0.69**	**0.81/0.76**	0.93/**0.89**	**0.78/0.80**	**0.87/0.81**	0.93/0.90	**0.93/0.86**

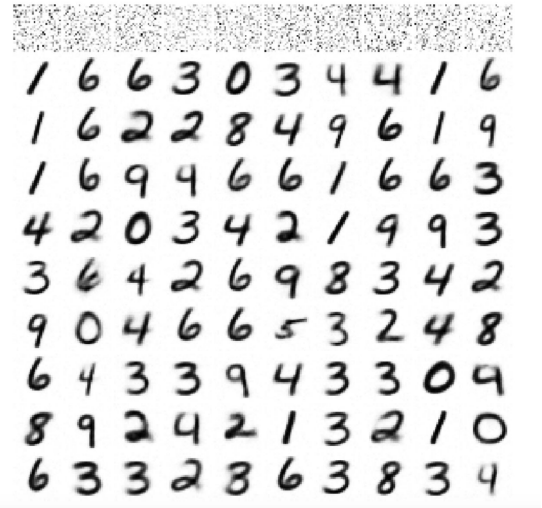
Extremely fast decoding

- Amortize the decoding cost
  - Most probable decoding obtained in one forward pass through the network
- Up to 2x orders of magnitude in speedup on GPU vs. LDPC decoder



Generative Modeling Perspective

- Becomes an implicit generative model
- Discrete version of the Uncertainty Autoencoder (UAE)*
  - Variant of VAE without KL penalty term
- Parallels to VAE, Beta-VAE, Denoising Autoencoders



*Grover and Ermon, Variational compressive sensing using uncertainty autoencoders. UAI Workshop 2018

Summary

- End-to-end deep generative modeling framework for the JSCC problem
- Get much better bitlength efficiency
- Learns robust latent representations
- Get an extremely fast decoder for free

Learning Fair and Controllable Representations

Jiaming Song

w/ Ria Kalluri, Aditya Grover, Shengjia Zhao, Stefano Ermon

Stanford University

Problem setup

- X: features of an individual
- U: sensitive attribute (e.g. gender)
- C = c(X, U) predicted values
- Y: target variable (labels)
- Z: representation (used for downstream tasks)

Make accurate predictions while protecting U.

- “Give loan according to credit but fair to race”

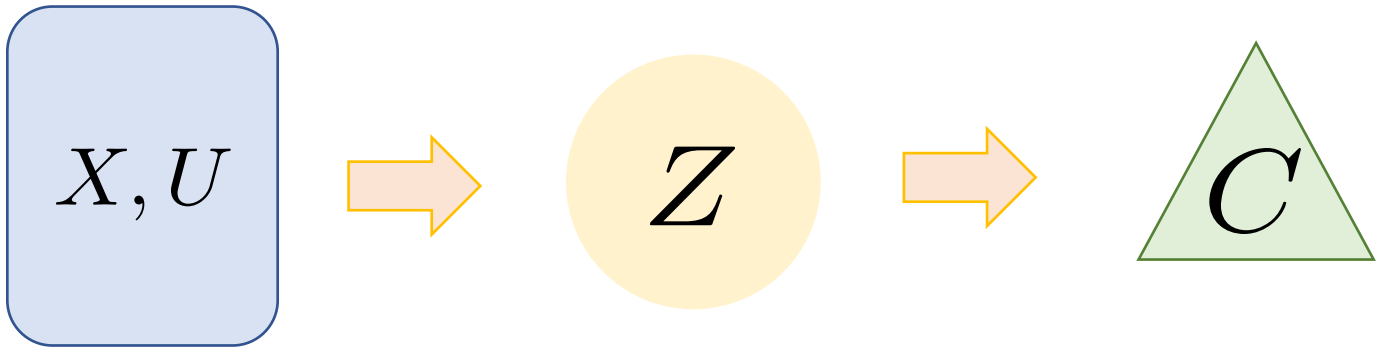
Examples of Fairness Notions

- Demographic parity
  - C and U are independent.
  - I(C; U) = 0
- Equalized odds
  - C and U are independent conditional on Y
  - I(C; U|Y) = 0
- Equalized opportunity
  - C and U are independent conditional on Y = 1
  - I(C; U|Y=1) = 0

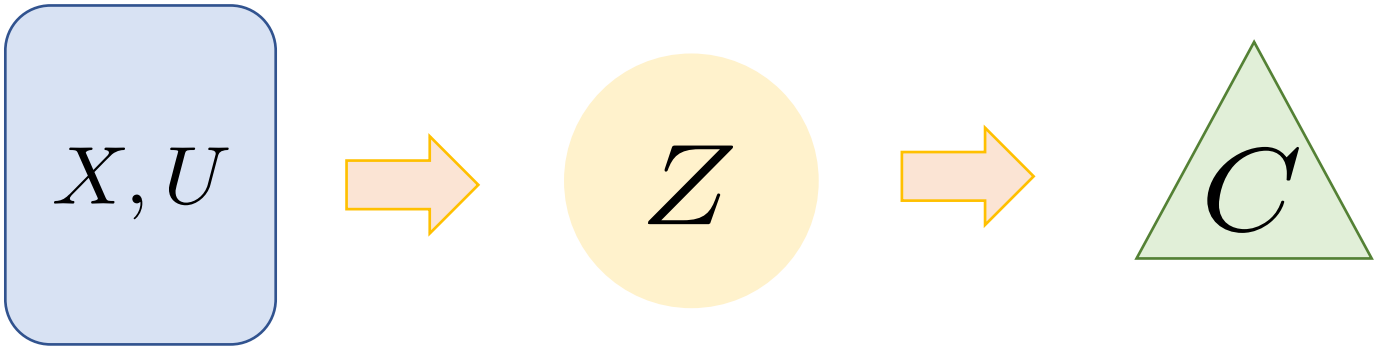
Not all notions can be satisfied at once!

Representation Learning

- Learn a representation Z, and use Z to predict Y
- “Data Preprocessing”



Learning Fair and Expressive Representations



$\max I(X; Z|U)$ $\min I(U; Z)$

Expressiveness Fairness

Optimization Problem

- Maximize “expressiveness”
- Under “fairness” constraints
- Prediction model $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{u})$
- Data distribution $q(\mathbf{x},\mathbf{u})$
- Constrained optimization problem
  - e.g. demographic parity

$$\max I_q(\mathbf{x};\mathbf{z}|\mathbf{u})$$
$$\text{s.t.} \quad I_q(\mathbf{z};\mathbf{u}) < \epsilon$$

Quantity determined by user (hyperparameter)

Tractable Bounds for Mutual Information

These quantities are not “tractable”!

$$I_q(\mathbf{x};\mathbf{z}|\mathbf{u}) = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z},\mathbf{u})}[\log q_\phi(\mathbf{x},\mathbf{z}|\mathbf{u}) - \log q(\mathbf{x}|\mathbf{u}) - \log q_\phi(\mathbf{z}|\mathbf{u})]$$

$$I_q(\mathbf{z};\mathbf{u}) = \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{u})}[\log q_\phi(\mathbf{z}|\mathbf{u}) - \log q_\phi(\mathbf{z})]$$

Only $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{u})$ has tractable log density!

$$\max I_q(\mathbf{x}; \mathbf{z}|\mathbf{u})$$

- Introduce a parametrized distribution $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{u})$

$$I_q(\mathbf{x}; \mathbf{z}|\mathbf{u}) = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z},\mathbf{u})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{u})] + H_q(\mathbf{x}|\mathbf{u}) + \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{u})} D_{\mathrm{KL}}(q_\phi(\mathbf{x}|\mathbf{z}, \mathbf{u}) \| p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{u}))$$

"Reconstruction Error"
"Distortion"

"Constant Entropy"

KL divergence > 0

Use "Reconstruction Error" as lower bound (up to constant)!

Fixing a Broken ELBO, Alemi et al. ICML 2018.

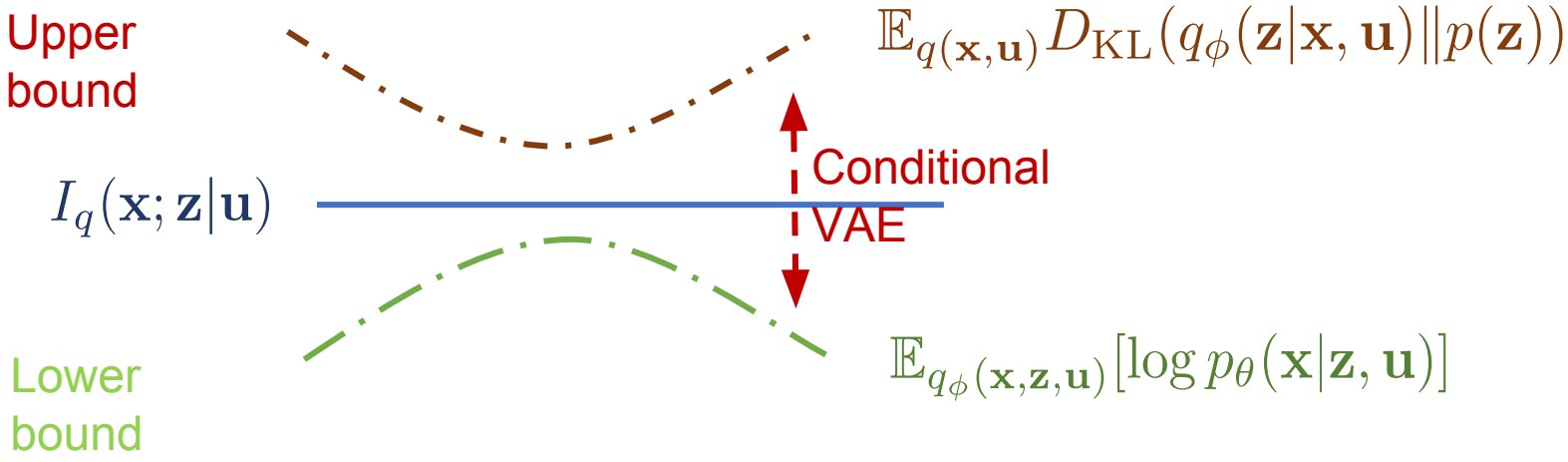
$$\min I_q(\mathbf{u}, \mathbf{z})$$

$$I_q(\mathbf{z}; \mathbf{u}) \leq I_q(\mathbf{z}; \mathbf{x}, \mathbf{u}) = \mathbb{E}_{q(\mathbf{x},\mathbf{u})} D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{u}) \| p(\mathbf{z})) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}) \| p(\mathbf{z}))$$

Add additional variable

"Rate" Upper Bound

KL divergence > 0



A Lagrangian Perspective of Latent Variable Generative Models, Zhao, Song, Ermon. UAI 2018.

$$\min I_q(\mathbf{u}, \mathbf{z})$$

The “rate” upper bound is not tight!

Tighter version with some $p(\mathbf{u})$:

$$I_q(\mathbf{z}; \mathbf{u}) = \mathbb{E}_{q_\phi(\mathbf{z})} D_{\mathrm{KL}}(q_\phi(\mathbf{u}|\mathbf{z}) \| p(\mathbf{u})) - D_{\mathrm{KL}}(q(\mathbf{u}) \| p(\mathbf{u}))$$

Upper bound, but we don’t know $q_\phi(\mathbf{u}|\mathbf{z})$

Approximate $q_\phi(\mathbf{u}|\mathbf{z})$ with $p_\psi(\mathbf{u}|\mathbf{z})$ by maximum likelihood!

“Adversarial Training”

Approximate $q_\phi(\mathbf{u}|\mathbf{z})$ with $p_\psi(\mathbf{u}|\mathbf{z})$ by maximum likelihood!

$$\min_\psi \mathbb{E}_{q_\phi(\mathbf{z})} D_{\mathrm{KL}}(q_\phi(\mathbf{u}|\mathbf{z}) \| p_\psi(\mathbf{u}|\mathbf{z}))$$

Replace q with p, we have

$$\mathbb{E}_{q_\phi(\mathbf{z},\mathbf{u})}[\log p_\psi(\mathbf{u}|\mathbf{z}) - \log p(\mathbf{u})]$$

“Lower bound to upper bound”

$$= \mathbb{E}_{q_\phi(\mathbf{z})}[D_{\mathrm{KL}}(q_\phi(\mathbf{u}|\mathbf{z}) \| p(\mathbf{u})) - D_{\mathrm{KL}}(q_\phi(\mathbf{u}|\mathbf{z}) \| p_\psi(\mathbf{u}|\mathbf{z}))]$$

$$\leq \mathbb{E}_{q_\phi(\mathbf{z})} D_{\mathrm{KL}}(q_\phi(\mathbf{u}|\mathbf{z}) \| p(\mathbf{u}))$$

“Maximum likelihood”

Maximum likelihood --> Gap with upper bound = 0!

“Adversarial Training” Intuition

Classifier: predict u from z.

Representation: prevent classifier to predict u.

- Adversarial training but not a GAN!
- Allows for any type of u (as opposed to binary)

“Tractable” Objective

$$\min_{\theta,\phi} \max_{\psi \in \Psi} \quad \mathcal{L}_r = -\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z},\mathbf{u})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{u})]$$

$$\text{s.t.} \quad C_1 = \mathbb{E}_{q(\mathbf{x},\mathbf{u})} D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{u}) \| p(\mathbf{z})) < \epsilon_1$$

$$C_2 = \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{u})}[\log p_\psi(\mathbf{u}|\mathbf{z}) - \log p(\mathbf{u})] < \epsilon_2$$

L: lower bound to “expressiveness”
C1, C2: upper bounds to “fairness”

Dual Optimization

Straightforward to use dual optimization

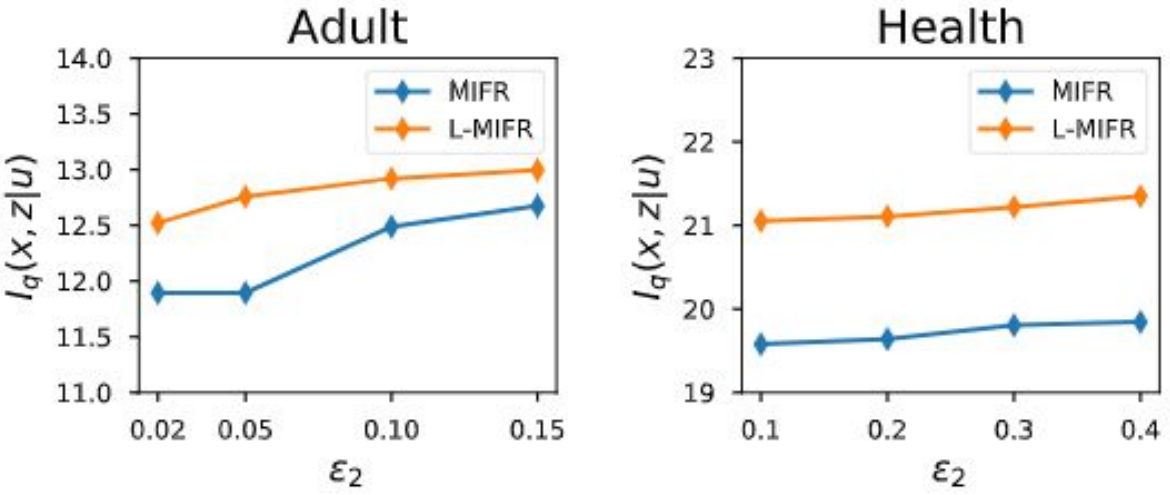
$$\max_{\lambda_1,\lambda_2 \geq 0} \min_{\theta,\phi} \max_{\psi} \mathcal{L} = \mathcal{L}_r + \lambda_1^\top (C_1 - \epsilon_1) + \lambda_2^\top (C_2 - \epsilon_2)$$

If constraint is violated, increase its weight

- Find the trade-off between fairness and expressiveness!
- Particularly useful with multiple fairness notions!
- “Find solution that is reasonable under multiple notions”
- Allows user to control level of “unfairness” directly

Learning Better Representations

- Search for most expressive representations under certain fairness constraints.
- MIFR: fixed multipliers, grid search over 5x5 hyperparameters, find most “expressive” feasible solution.
- L-MIFR: train Lagrange multipliers on-the-fly, run once.



Learning Better Representations

- Search for most expressive representations under multiple fairness constraints.
  - Demographic parity, Equalized odds, Equalized opportunity
- MIFR: greedy hyperparameter search, 12 runs.
- L-MIFR: train Lagrange multipliers on-the-fly, run once.

	$I_q(\mathbf{x};\mathbf{z}\|\mathbf{u})$	C_1	I_{DP}	I_{EO}	I_{EOpp}
MIFR	9.34	**9.39**	0.09	0.10	0.07
L-MIFR	**9.94**	9.95	**0.08**	**0.09**	**0.04**

L-MIFR learns better representations!

Summary

- Information-theoretic view of fair representation learning
- Connection with VAEs and GANs
- Dual optimization avoids tedious hyperparameter tuning and learns trade-offs
- Better representation with less compute

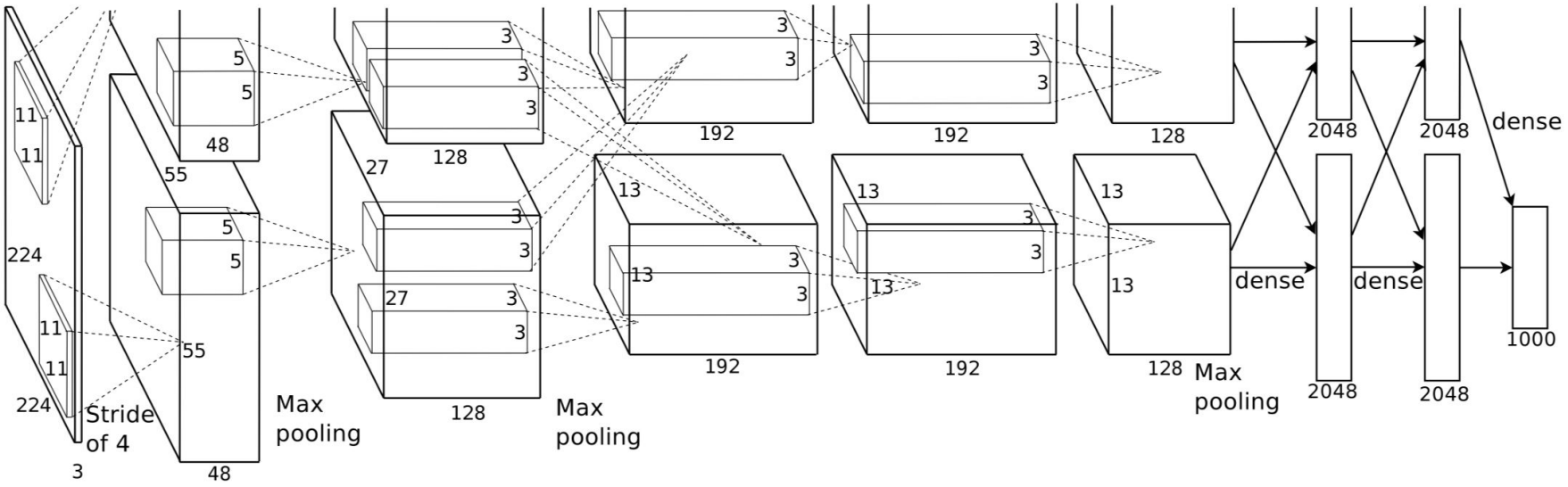
Constructing Unrestricted Adversarial Examples with Generative Models

Yang Song

w/ Rui Shu, Nate Kushman, Stefano Ermon

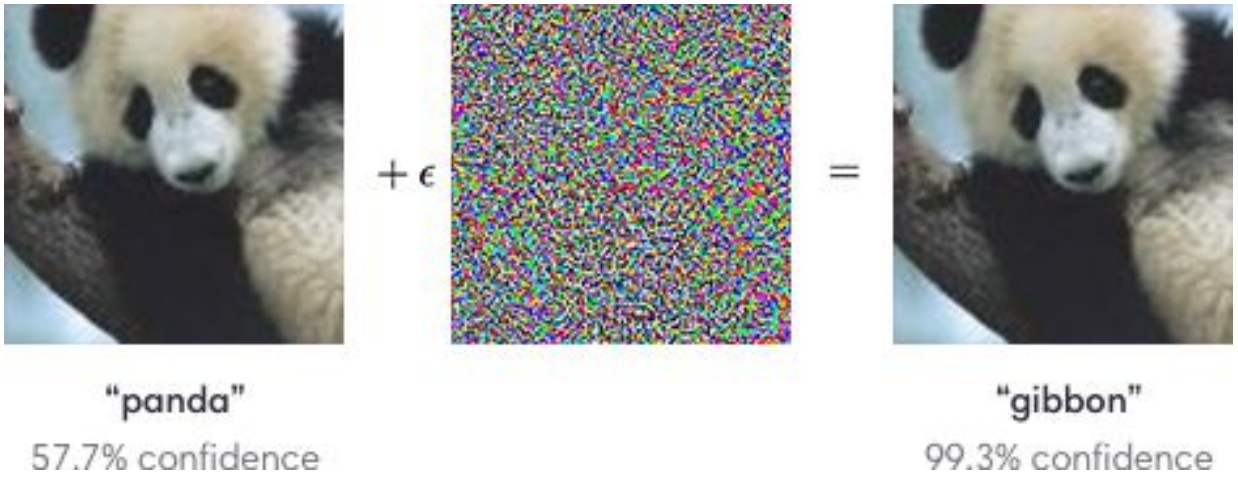
Stanford University

Success of Deep Learning



- Classifiers based on deep convolutional networks have surpassed human performance in many benchmarks.
- The superior predictive power of deep neural networks enables success beyond image classification, e.g., generative models, reinforcement learning.
- Do neural networks **understand** images like humans do?

Adversarial Examples

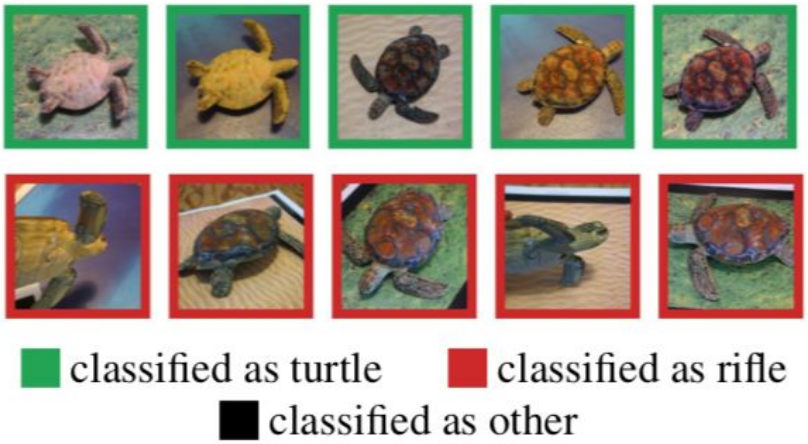


- Images perturbed by imperceptible noise can fool classifier.
- Construct adversarial examples via optimization:

$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \epsilon \operatorname{sign}\left(\nabla_X J(\boldsymbol{X}, y_{true})\right)$$

Why Care about Adversarial Examples

- Discloses the difference from human perception.
- Significant security vulnerability of machine learning.
  - Physical existence



Defenses against Adversarial Examples

- Adversarial training.
  - Generate adversarial examples on-the-fly and use them to augment the training dataset.
  - Not principled.
- Provable defenses.
  - Minimize the upper bound on worst loss for all perturbations.
  - Achieve more than 94% accuracy on MNIST for

    $\epsilon \leq 0.1$
- Is this the correct way to solving adversarial examples?

Unrestricted Adversarial Examples

- Adversarial examples are restricted to small norm-bounded noise.
- Small perturbation is not necessary for being a security vulnerability.

Unrestricted Adversarial Examples

- Unrestricted adversarial examples are legitimate images that miscategorize a classifier according to an oracle.

- No restriction of small noise and no need of test images.

Norm-constrained adversarial examples

Unrestricted adversarial examples

Constructing Unrestricted Adversarial Examples

- Perturbation-based adversarial examples preserve legitimacy by limiting the magnitude of noise.
- We propose to use a **conditional generative model** to approximately capture the notion of legitimacy.
- Search the latent space of a generative model so that it produces an adversarial example.
- Those unrestricted adversarial examples are named *Generative Adversarial Examples*.

Generative Adversarial Examples

$$x = g_\theta(y, z), \quad z \sim p(Z)$$

$$f(g_\theta(y_{source}, z')) = y_{target} \neq y_{source}$$

Classifier

Conditional generative model

The label that f is fooled to predict.

The label that the image truly belongs to

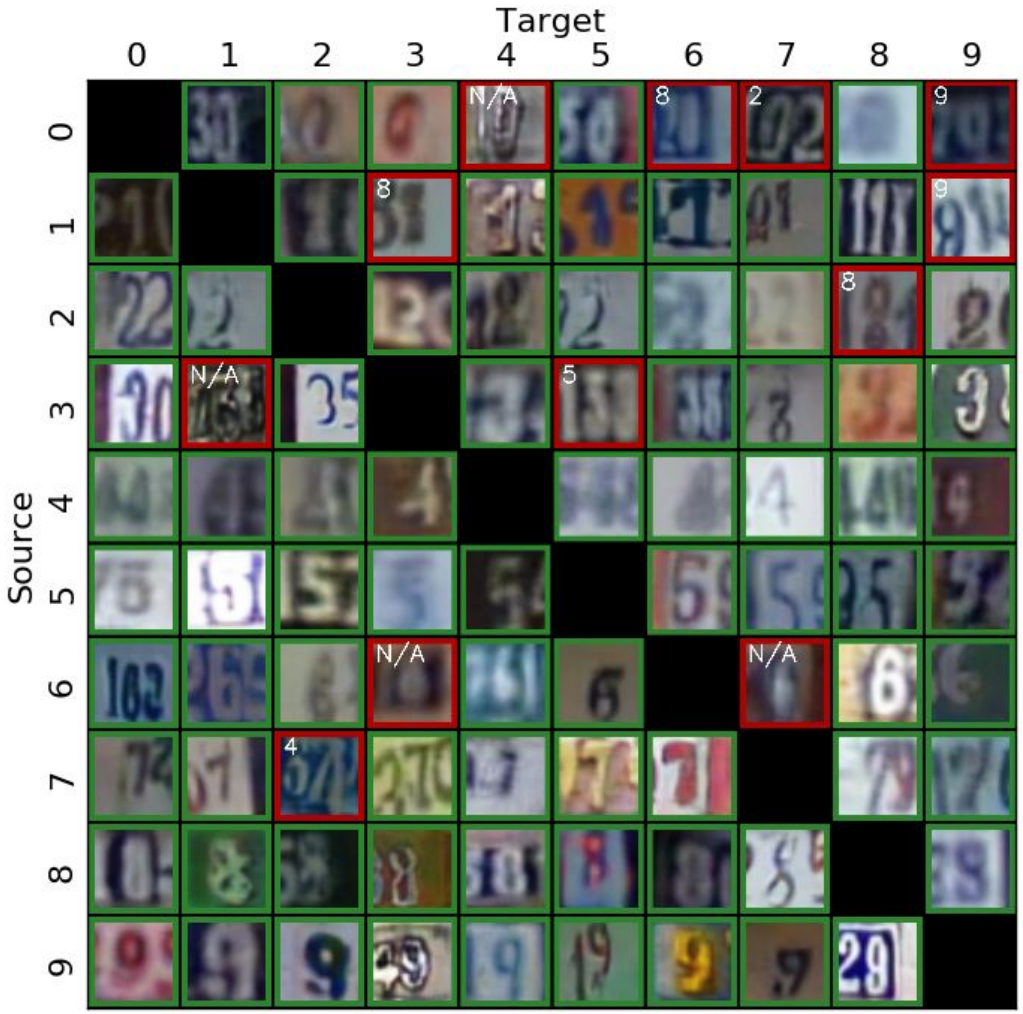
Generative Adversarial Examples
MNIST

Generative Adversarial Examples
MNIST

Table 1: Attacking certified defenses on MNIST. The generative adversarial examples here are untargeted and without noise-augmentation. Numbers represent success rates (%) of our attack, based on human evaluations on MTurk. No perturbation-based attack with $\epsilon = 0.1$ can have a success rate larger than the *certified rate*, when evaluated on the training set.

Classifier \ Source	0	1	2	3	4	5	6	7	8	9	Certified Rate ($\epsilon = 0.1$)	Overall
Raghunathan et al. [16]	90.8	48.3	86.7	93.7	94.7	85.7	93.4	80.8	96.8	95.0	35.0	86.6
Kolter & Wang [17]	94.2	57.3	92.2	94.0	93.7	89.6	95.7	81.4	96.3	93.5	5.8	88.8

Generative Adversarial Examples SVHN

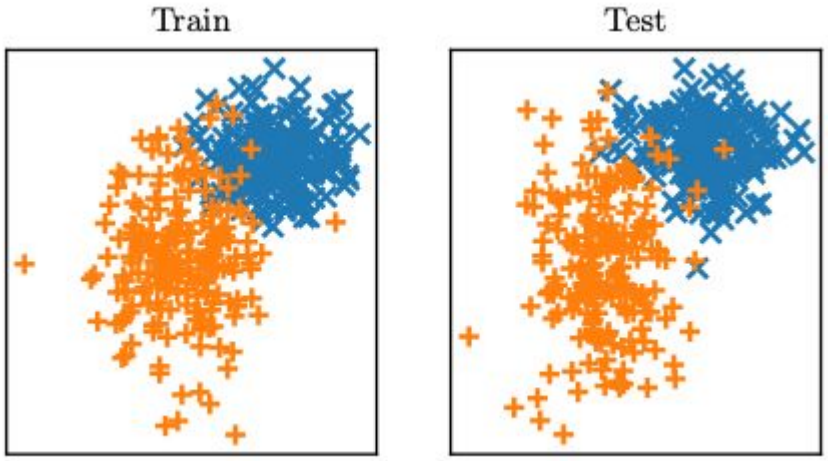


Generative Adversarial Examples
CelebA - female

Generative Adversarial Examples
CelebA - male

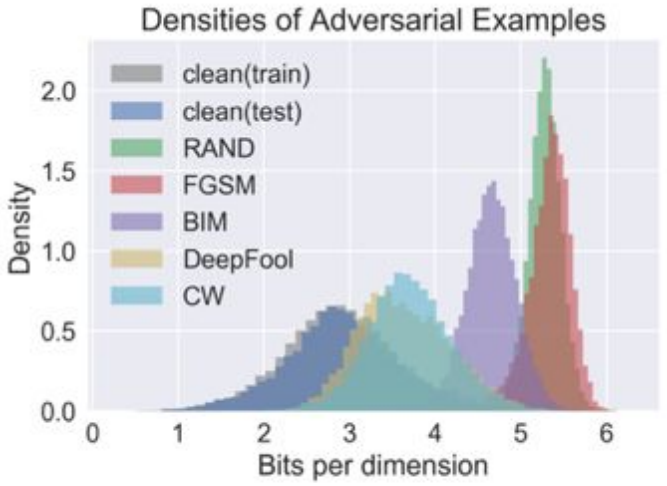
i.i.d Machine Learning

- Training and test examples are drawn **independently** from the **same** data-generating distribution.
- Evaluation is based on **expected risk** on the shared data-generating distribution.



Caveats to the i.i.d framework

- Test data can be from another distribution.
  - Adversarial examples
  - Unusual inputs



(Song et al, 2018)

- Test data can be dependent
  - By optimizing over z, generated images are not iid, in contrast to sampling z randomly.

Summary

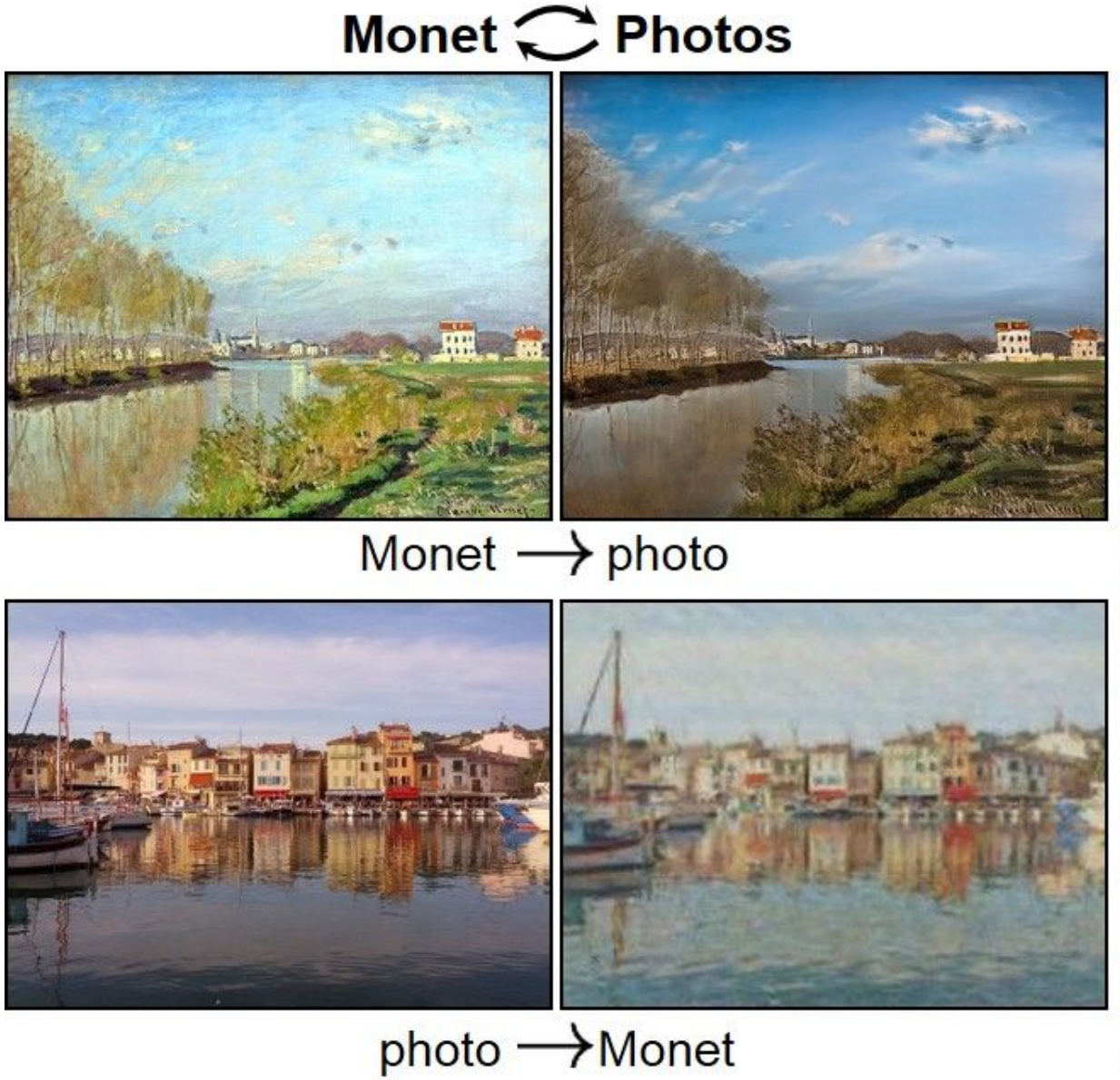
- Adversarial examples beyond norm-constrained perturbations exist and are easy to construct.
- We need to rethink defense methods focused on improving robustness against small perturbations.
- We need to rethink the ways of approaching reliable machine learning. Research on covariate shift, and reliable uncertainty quantification could be useful.

CycleGAN, a Master of Steganography

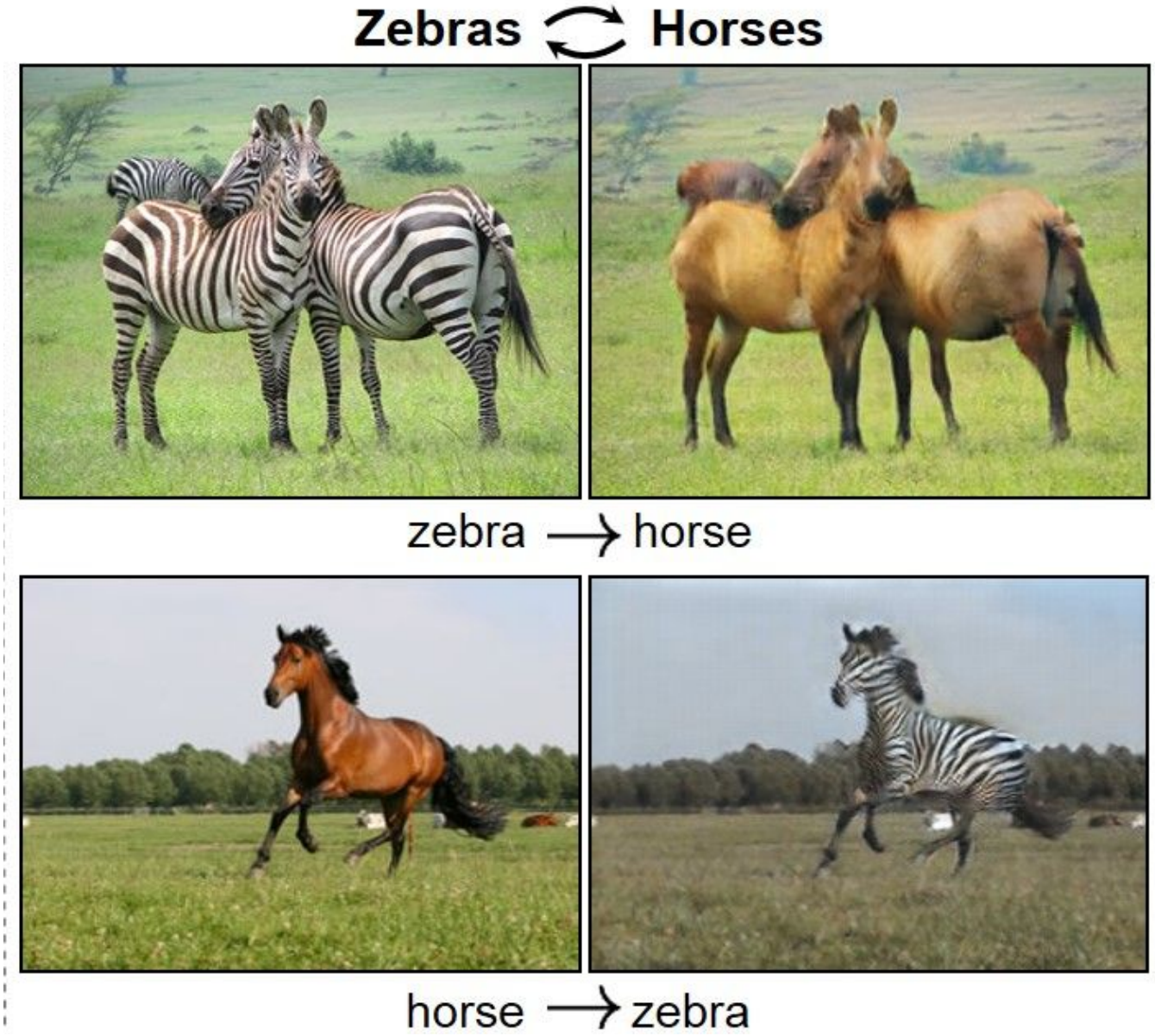
Casey Chu

w/ Andrey Zhmoginov and Mark Sandler

Stanford University



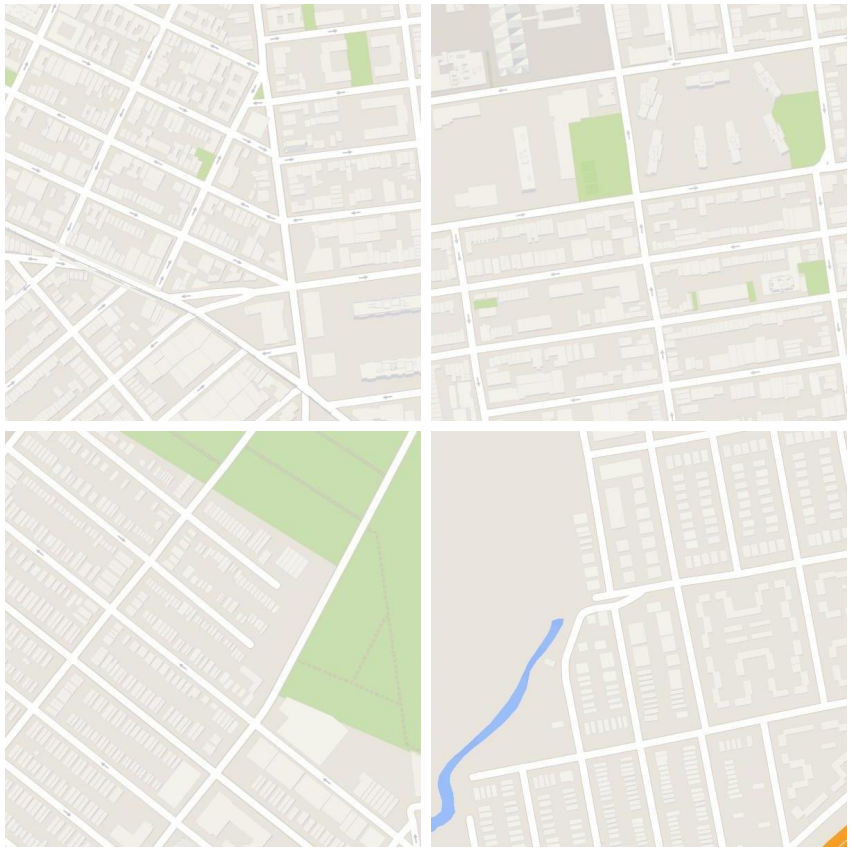
Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. “Unpaired image-to-image translation using cycle-consistent adversarial networks.” (2017).



Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. “Unpaired image-to-image translation using cycle-consistent adversarial networks.” (2017).

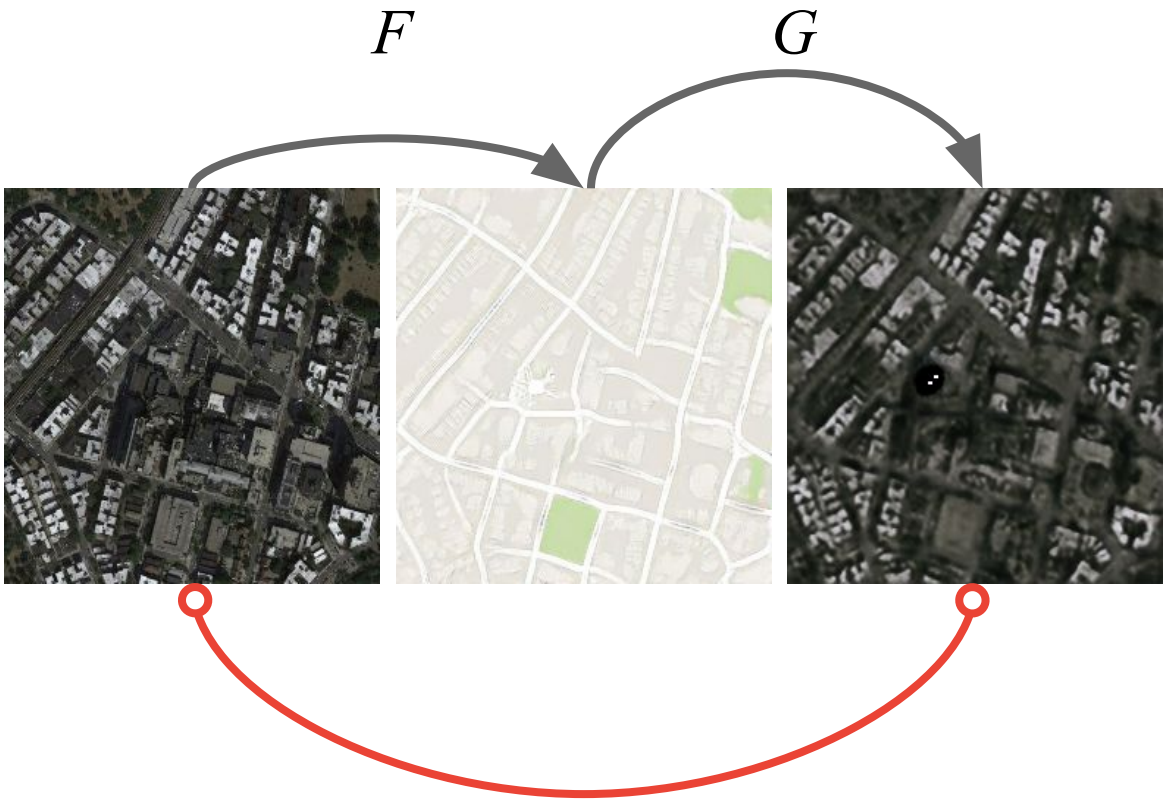
Domain A: **aerial imagery**

Domain B: **maps**

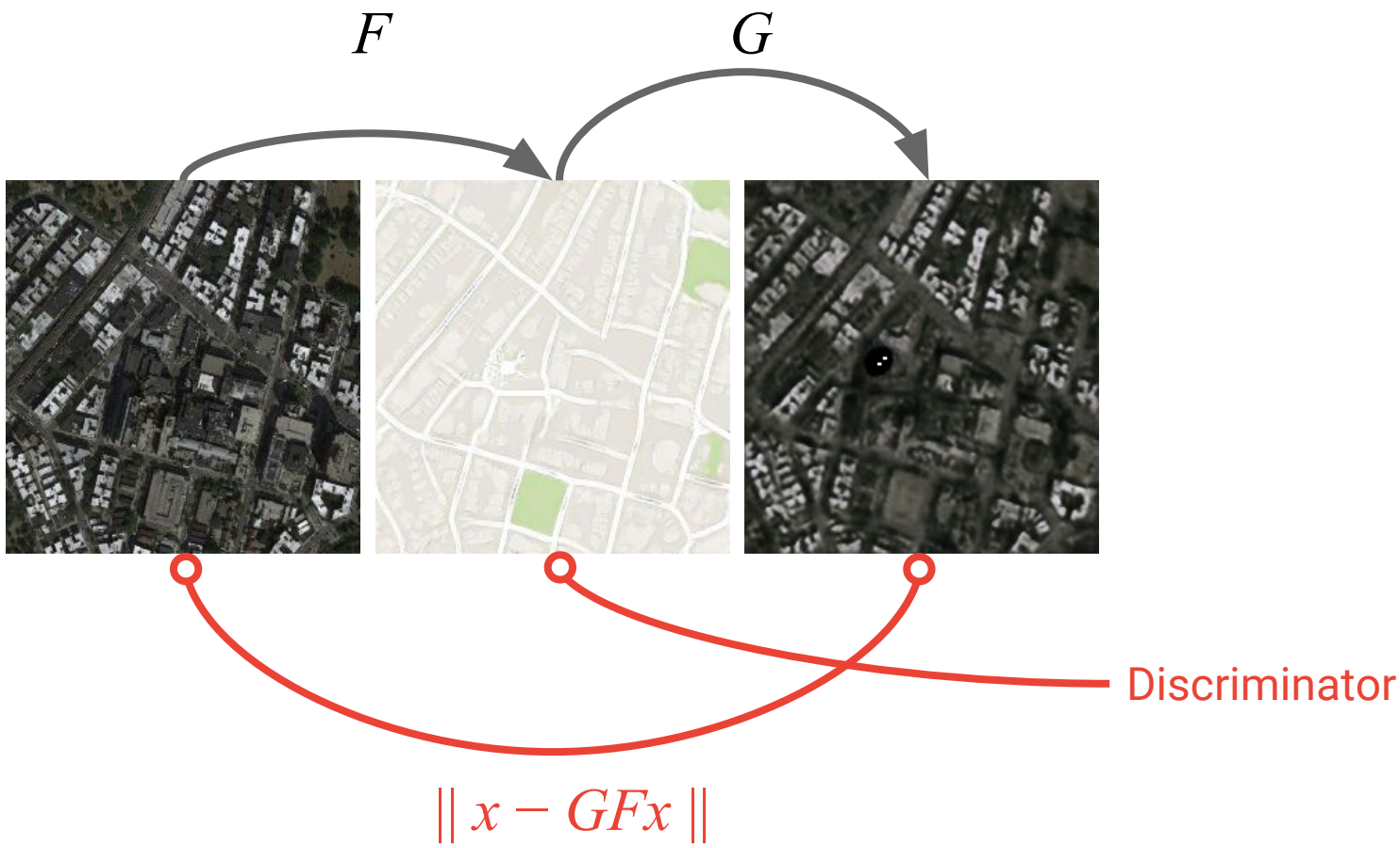


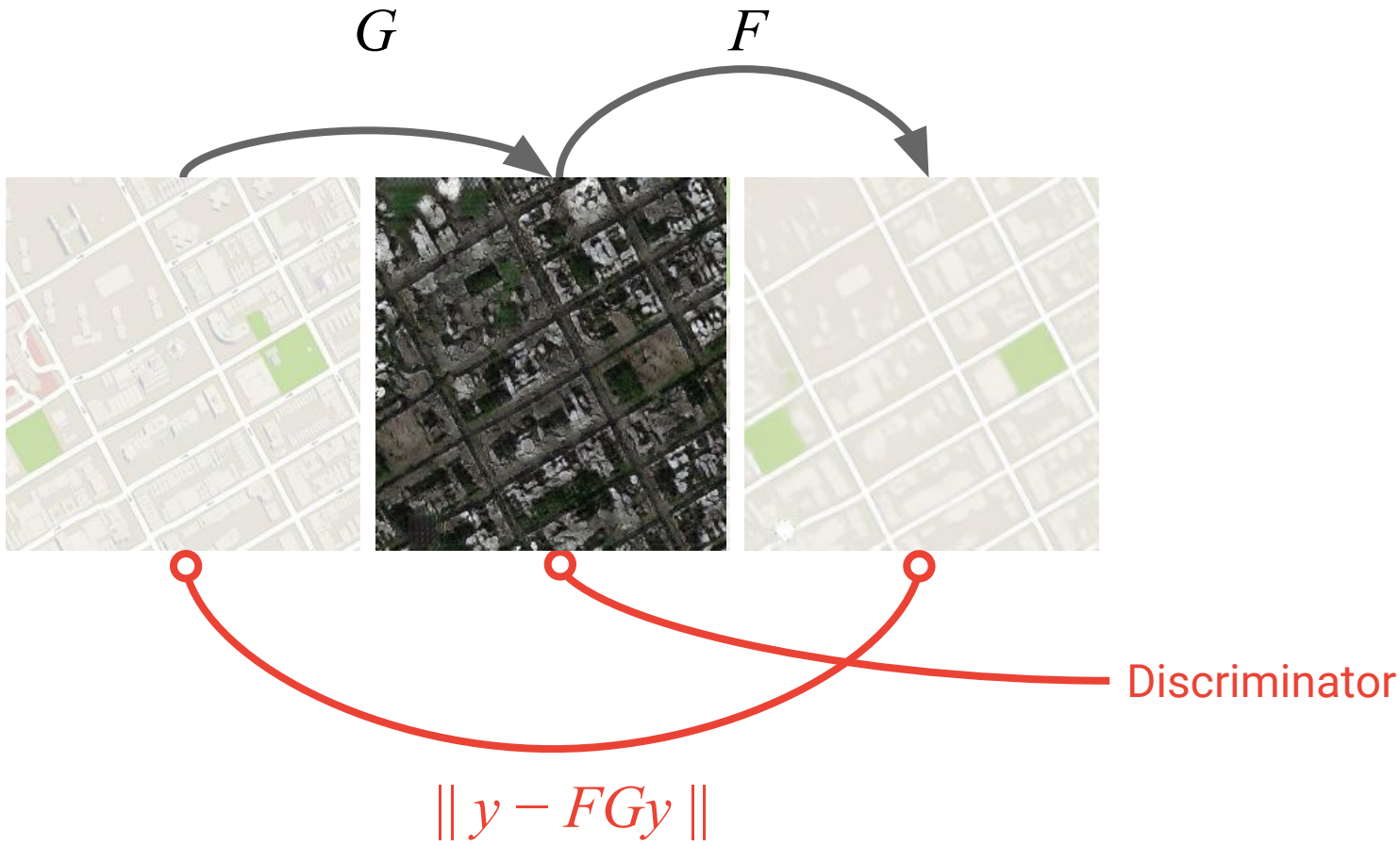
F

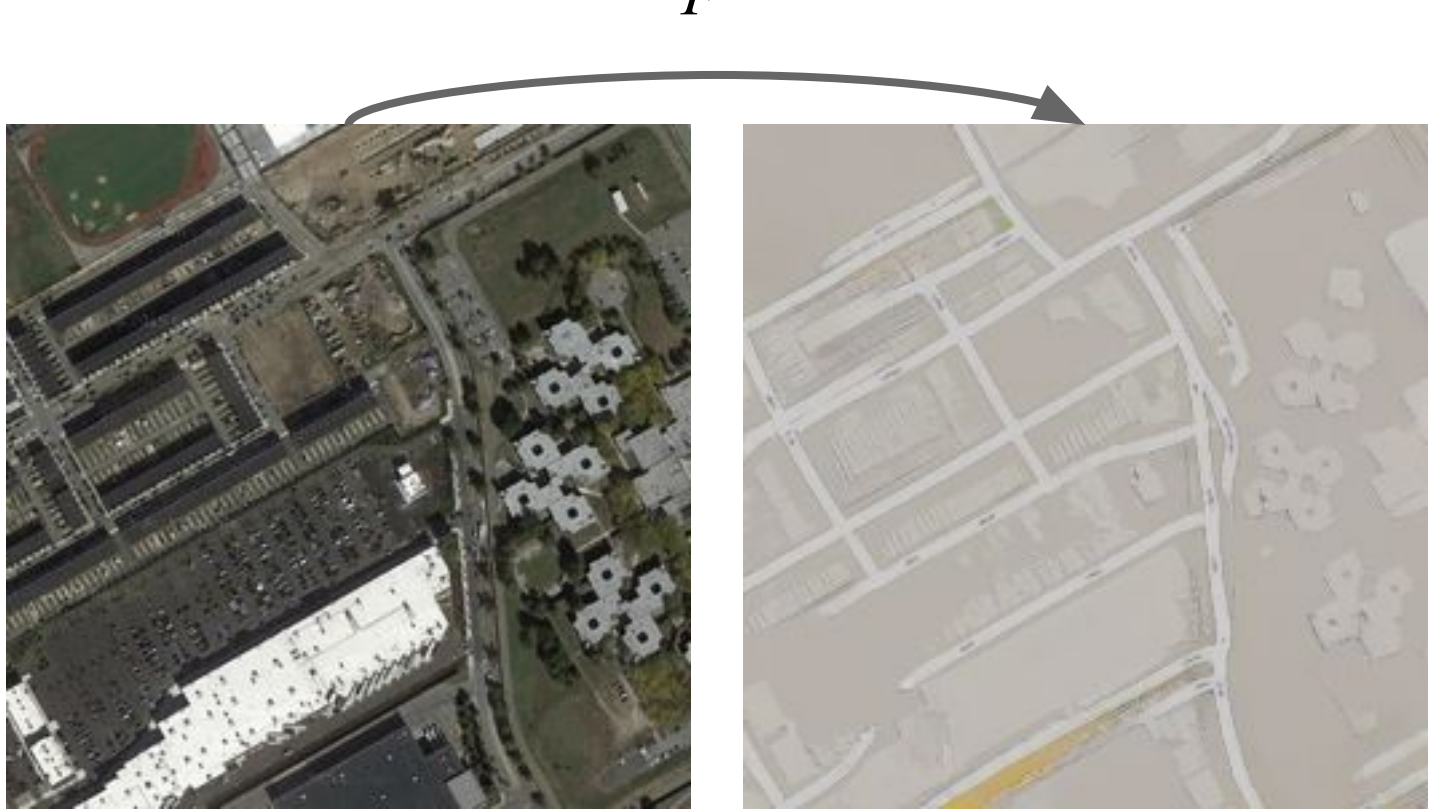

F
G



$|| x - GFx ||$ Cyclic consistency loss

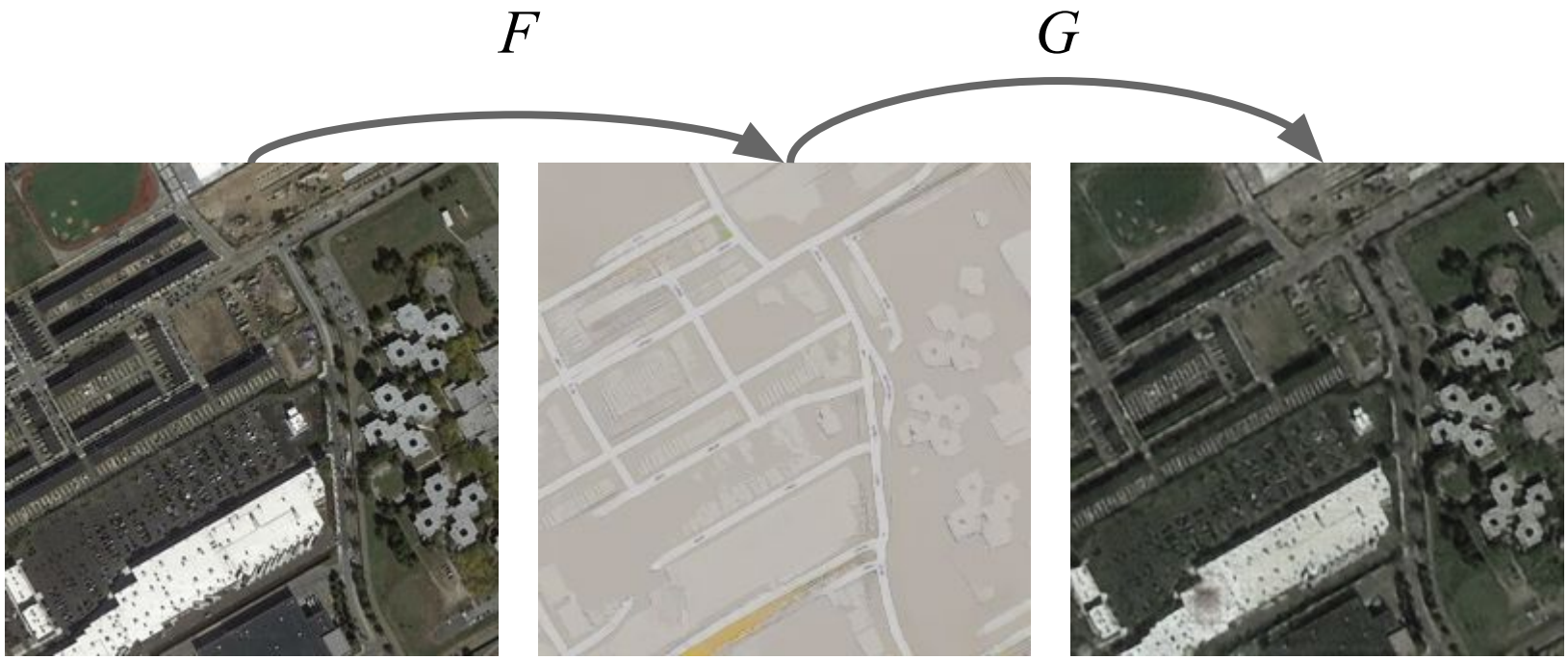

F
G
Discriminator
|| x − GFx ||

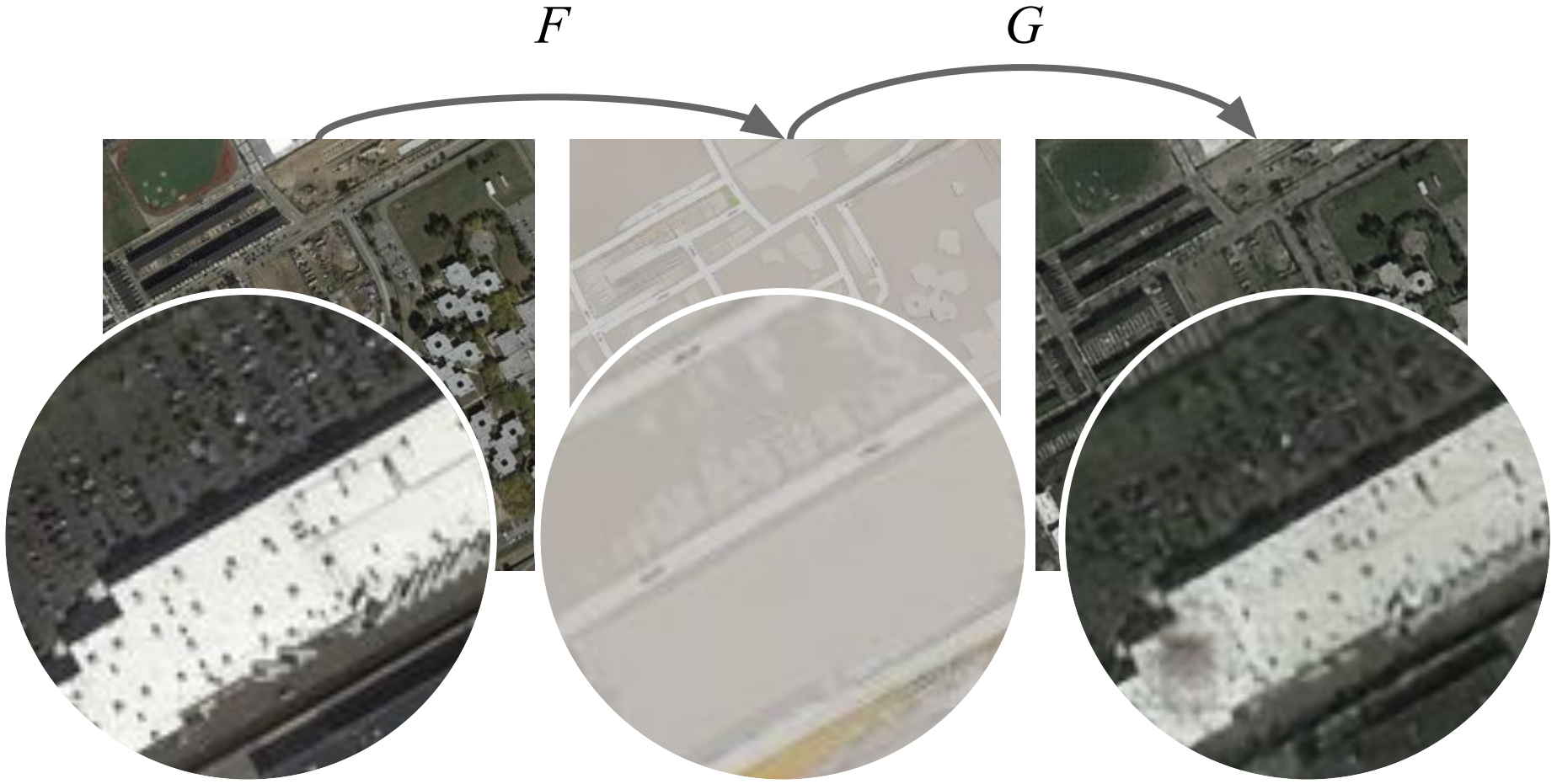

G
F
Discriminator
|| y − FGy ||

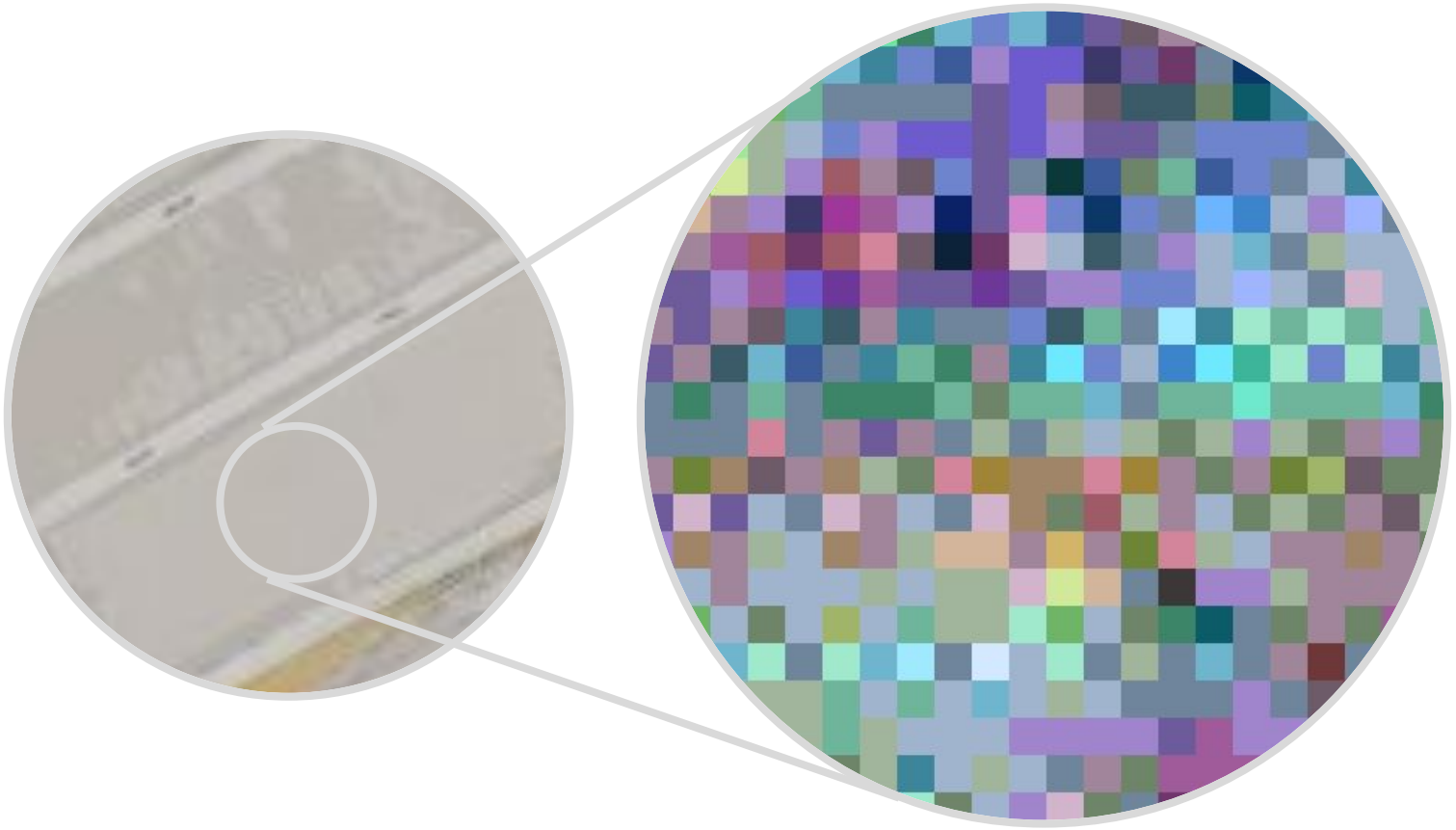

F

G

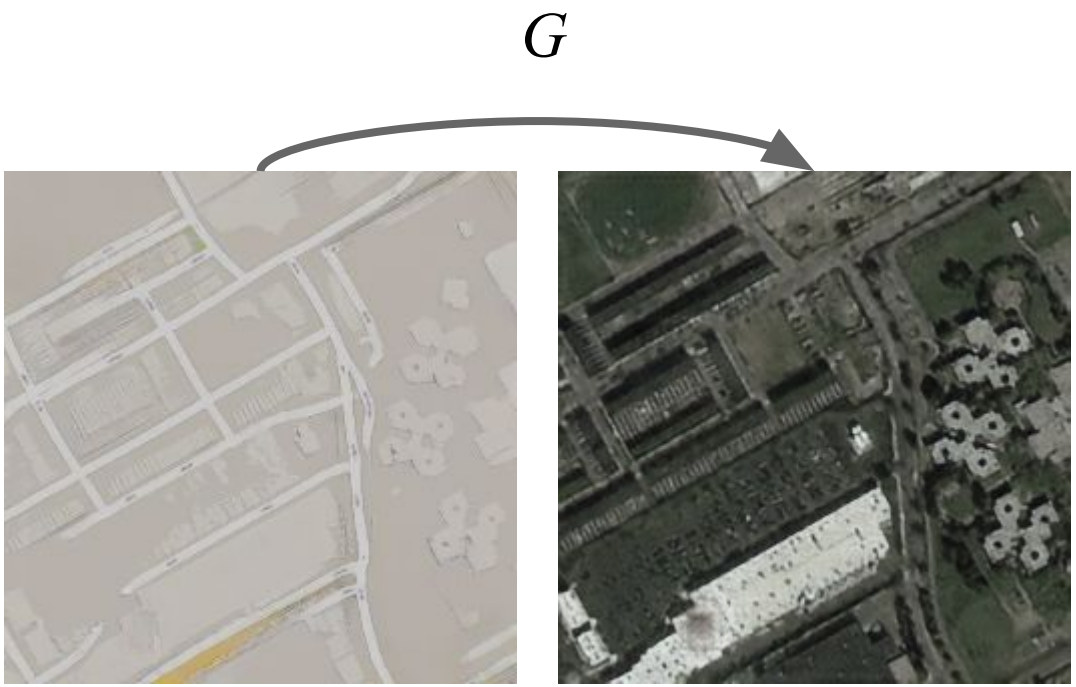
This architecture works very well… *too* well?

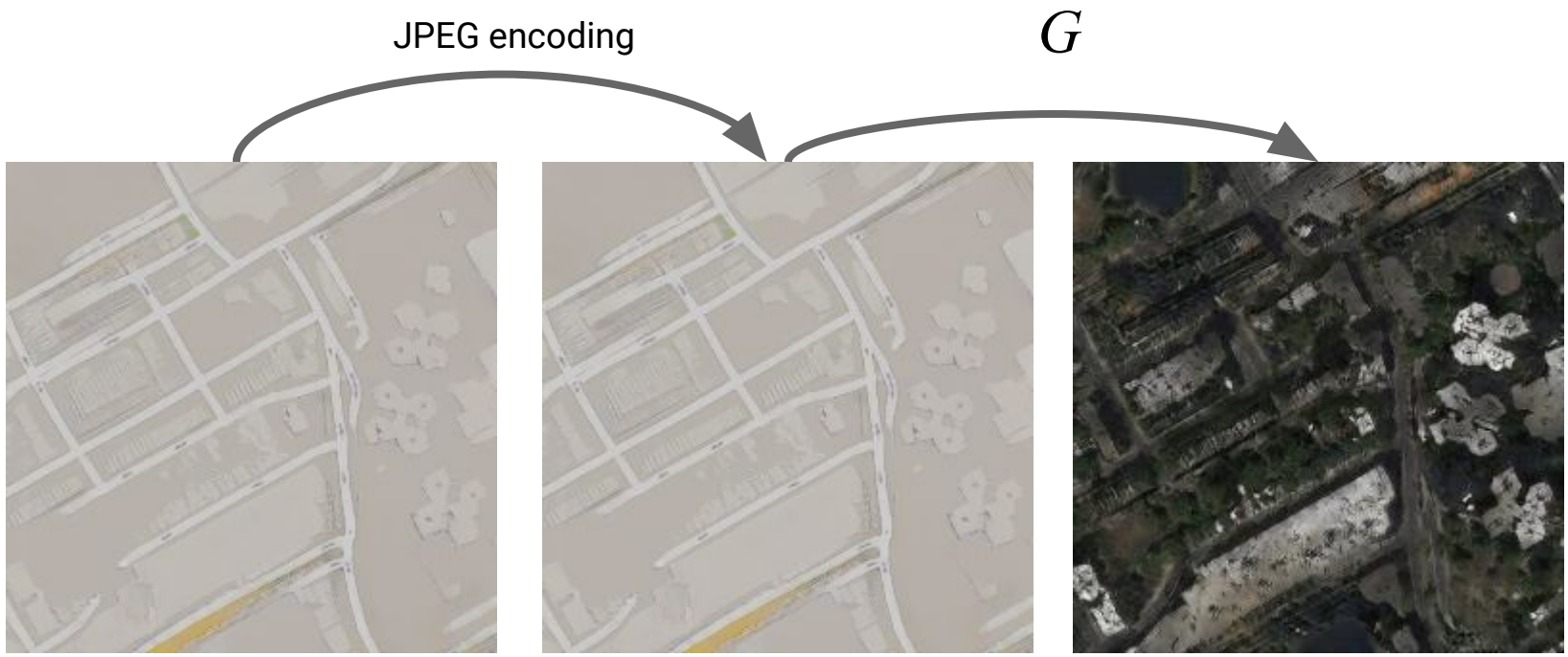

F
G


F
G



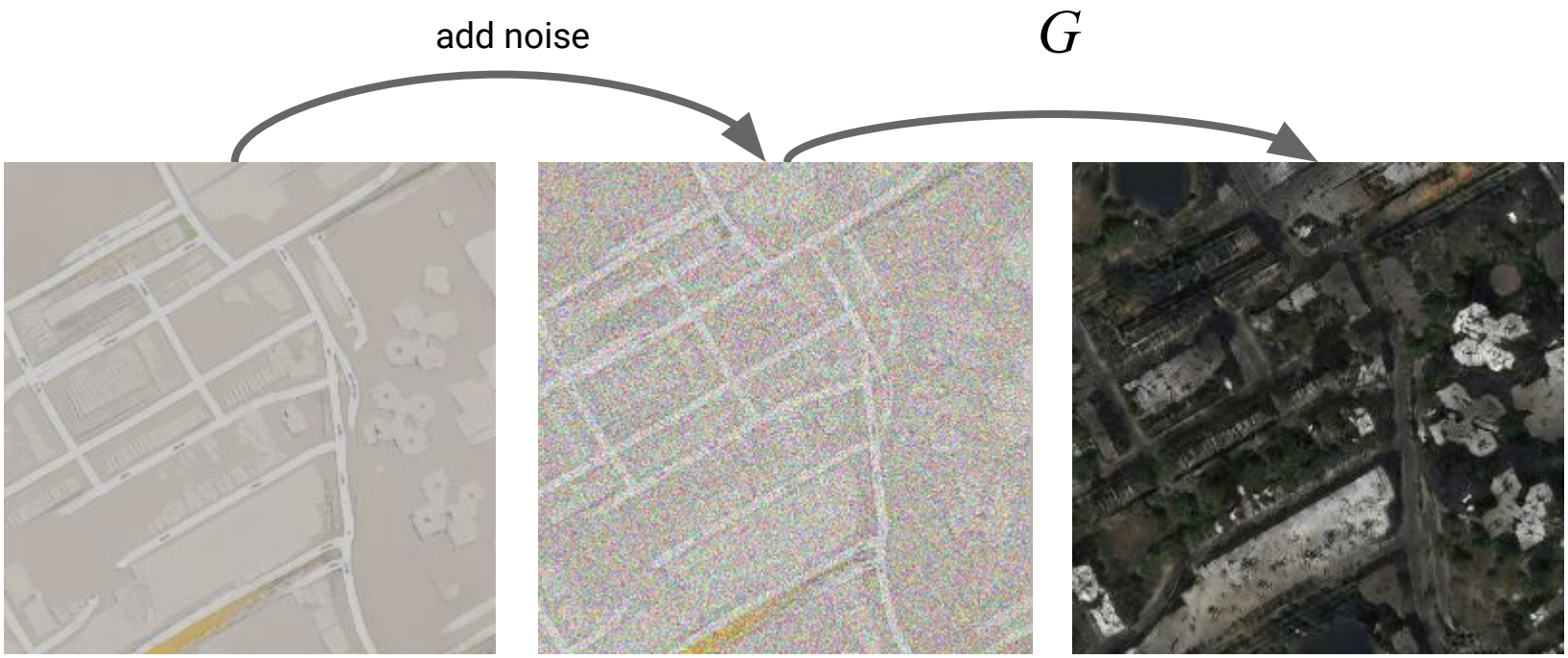
The cyclic consistency loss is asking for too much.


G


JPEG encoding
G

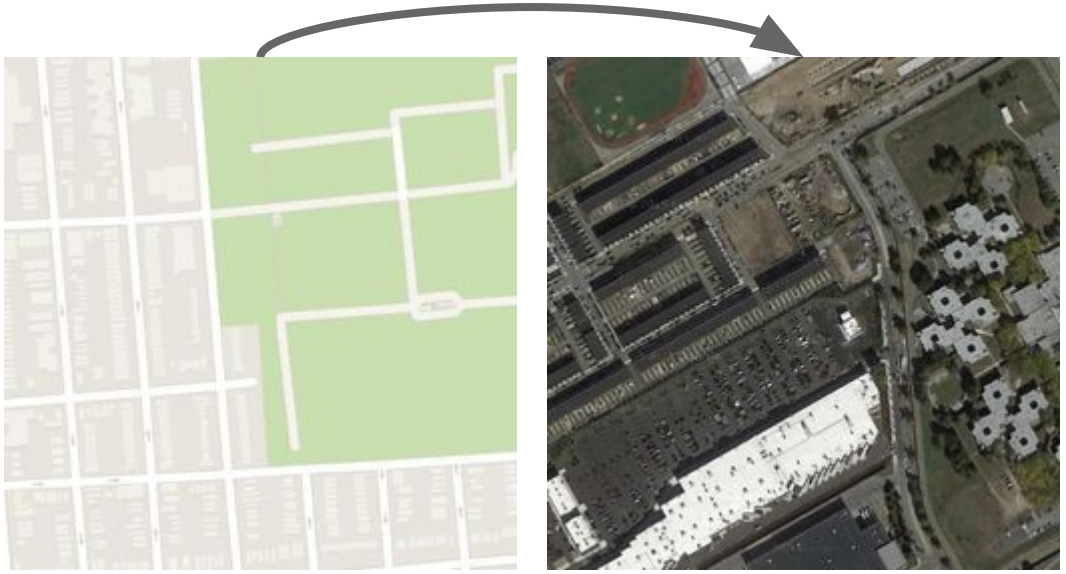
no encoding

with JPEG encoding


add noise
G

CycleGAN stores information within its generated images in a high-frequency, low-amplitude signal.

G?



Can we hide an arbitrary aerial image in an arbitrary map?

y_0 x

$$y^* = \arg\min_y \| Gy - x \|$$

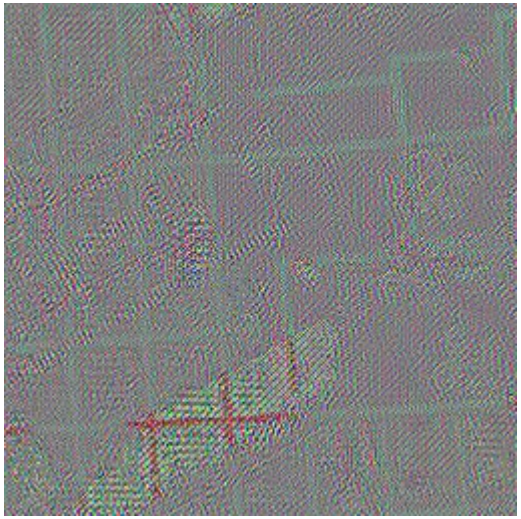
(Similar to Szegedy et al.'s "Intriguing properties of neural networks.")


G
y^*
Gy^*
x

y_0

y^*

$y_0 - y^*$



(amplified for visibility)

Adversarial attack:

$$y^* = \arg\min_y \| Gy - x \|$$

CycleGAN training criterion:

$$F, G = \arg\min_{F,\,G} \| GFx - x \| + \text{other terms}$$

Because of its cyclic consistency loss, CycleGAN is, effectively, attacking itself.