Using pre-trained generative models as priors for image reconstruction problems.

Arthur Conmy and Subhadip Mukherjee

Cambridge Image Analysis Seminar, May 2021

UNIVERSITY OF CAMBRIDGE

Plan

1. A brief history of GANs.

2. StyleGAN and the generative state-of-the-art.

3. Inversion, reconstruction and current work.

UNIVERSITY OF CAMBRIDGE

Plan

1. A brief history of GANs.
2. StyleGAN and the generative state-of-the-art.
3. Inversion, reconstruction and current work.

Where GANs came from.

The original motivation for GANs came from a game theoretic standpoint; pit two neural networks G and D against each other and define the natural analogue of cross entropy loss in this case:

$$V(D, G) = \mathbb{E}_{x \sim P_r}[\log D(x)] + \mathbb{E}_{z \sim \mathcal{N}_n(0,I)}[\log(1 - D(G(z)))] \tag{1}$$

(the distribution of generated images, $G(z)$ will be denoted P_g).

- We're already in the setting to apply back-prop, so what's wrong?

UNIVERSITY OF CAMBRIDGE

Where GANs came from.

The original motivation for GANs came from a game theoretic standpoint; pit two neural networks G and D against each other and define the natural analogue of cross entropy loss in this case:

$$V(D, G) = \mathbb{E}_{x \sim P_r}[\log D(x)] + \mathbb{E}_{z \sim \mathcal{N}_n(0,I)}[\log(1 - D(G(z)))] \qquad (1)$$

(the distribution of generated images, $G(z)$ will be denoted P_g).

- We're already in the setting to apply back-prop, so what's wrong?
- This miserably fails.

UNIVERSITY OF CAMBRIDGE

Where GANs came from.

The original motivation for GANs came from a game theoretic standpoint; pit two neural networks G and D against each other and define the natural analogue of cross entropy loss in this case:

$$V(D, G) = \mathbb{E}_{x \sim P_r}[\log D(x)] + \mathbb{E}_{z \sim \mathcal{N}_n(0,I)}[\log(1 - D(G(z)))] \qquad (1)$$

(the distribution of generated images, $G(z)$ will be denoted P_g).

- We're already in the setting to apply back-prop, so what's wrong?
- This miserably fails.
- More theoretical analysis leads to modifying the V above to fix the vanishing gradients problem.

UNIVERSITY OF CAMBRIDGE

Where GANs came from.

The original motivation for GANs came from a game theoretic standpoint; pit two neural networks G and D against each other and define the natural analogue of cross entropy loss in this case:

$$V(D, G) = \mathbb{E}_{x \sim P_r}[\log D(x)] + \mathbb{E}_{z \sim \mathcal{N}_n(0,I)}[\log(1 - D(G(z)))] \quad (1)$$

(the distribution of generated images, $G(z)$ will be denoted P_g).

- We're already in the setting to apply back-prop, so what's wrong?
- This miserably fails.
- More theoretical analysis leads to modifying the V above to fix the vanishing gradients problem.
- However, the training remains unstable, and highly dependent on heuristics and parameter tuning.

UNIVERSITY OF CAMBRIDGE

Wasserstein and theoretically principled GANs.

Reference: Arjovsky, Chintala and Bottou (2017) and Gulrajani et al (2017).

- Given the true distribution P_r and a generated distribution P_g, optimize
$$\mathcal{L}(p_r, p_g) \tag{2}$$
where $\mathcal{L}$ is some loss function between probability distributions.

UNIVERSITY OF CAMBRIDGE

Wasserstein and theoretically principled GANs.

Reference: Arjovsky, Chintala and Bottou (2017) and Gulrajani et al (2017).

- Given the true distribution P_r and a generated distribution P_g, optimize

$$\mathcal{L}(p_r, p_g) \tag{2}$$

  where $\mathcal{L}$ is some loss function between probability distributions.
  - $\mathcal{L}$ needs to be estimable from iid samples.

UNIVERSITY OF CAMBRIDGE

Wasserstein and theoretically principled GANs.

Reference: Arjovsky, Chintala and Bottou (2017) and Gulrajani et al (2017).

- Given the true distribution P_r and a generated distribution P_g, optimize

$$\mathcal{L}(p_r, p_g) \tag{2}$$

  where $\mathcal{L}$ is some loss function between probability distributions.
  - $\mathcal{L}$ needs to be estimable from iid samples.
  - $\mathcal{L}$ needs to be differentiable.

UNIVERSITY OF CAMBRIDGE

Wasserstein and theoretically principled GANs.

Reference: Arjovsky, Chintala and Bottou (2017) and Gulrajani et al (2017).

- Given the true distribution P_r and a generated distribution P_g, optimize

$$\mathcal{L}(p_r, p_g) \tag{2}$$

  where $\mathcal{L}$ is some loss function between probability distributions.
  - $\mathcal{L}$ needs to be estimable from iid samples.
  - $\mathcal{L}$ needs to be differentiable.
  - This leaves a lot of possiblities!

UNIVERSITY OF CAMBRIDGE

Wasserstein and theoretically principled GANs.

Reference: Arjovsky, Chintala and Bottou (2017) and Gulrajani et al (2017).

- Given the true distribution P_r and a generated distribution P_g, optimize

$$\mathcal{L}(p_r, p_g) \tag{2}$$

  where $\mathcal{L}$ is some loss function between probability distributions.
    - $\mathcal{L}$ needs to be estimable from iid samples.
    - $\mathcal{L}$ needs to be differentiable.
    - This leaves a lot of possiblities!
- The cross-entropy loss on the previous slide leads to GANs minimising the **Jensen-Shannon** divergence $\mathcal{L}_{JS}$ between the distributions. D_{KL} fixes vanishing gradients.

UNIVERSITY OF CAMBRIDGE

Wasserstein GANs

Reference:
`https://vincentherrmann.github.io/blog/wasserstein/` (great article).

- The Wasserstein distance between two *discrete* distributions is

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|. \quad (3)$$

UNIVERSITY OF CAMBRIDGE

Wasserstein GANs

Reference:
`https://vincentherrmann.github.io/blog/wasserstein/` (great article).

- The Wasserstein distance between two *discrete* distributions is

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|. \quad (3)$$

- This generalises to continuous distributions via a duality theorem:

$$\mathrm{EMD}(P_r, P_\theta) = \sup_{\|f\|_{L \leq 1}} \mathbb{E}_{x\sim P_r} f(x) - \mathbb{E}_{x\sim P_\theta} f(x). \quad (4)$$

UNIVERSITY OF CAMBRIDGE

Wasserstein GANs

Reference:
`https://vincentherrmann.github.io/blog/wasserstein/` (great article).

- The Wasserstein distance between two *discrete* distributions is

$$\mathrm{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|. \quad (3)$$

- This generalises to continuous distributions via a duality theorem:

$$\mathrm{EMD}(P_r, P_\theta) = \sup_{\|f\|_{L\leq 1}} \mathbb{E}_{x\sim P_r} f(x) - \mathbb{E}_{x\sim P_\theta} f(x). \quad (4)$$

- How do we model a complicated function class such as 1-Lipschitz functions? With neural nets of course!
(add $\mathbb{E}[(|\nabla f| - 1)^2]$ term to enforce $||f||_L \leq 1$).

UNIVERSITY OF CAMBRIDGE

Why Wasserstein?

As an explicit example, see the original Wasserstein paper!

Example 1 (Learning parallel lines). Let $Z \sim U[0,1]$ the uniform distribution on the unit interval. Let $\mathbb{P}_0$ be the distribution of $(0, Z) \in \mathbb{R}^2$ (a 0 on the x-axis and the random variable Z on the y-axis), uniform on a straight vertical line passing through the origin. Now let $g_\theta(z) = (\theta, z)$ with θ a single real parameter. It is easy to see that in this case,

- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$,
- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$
- $KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$
- and $\delta(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} 1 & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ . \end{cases}$

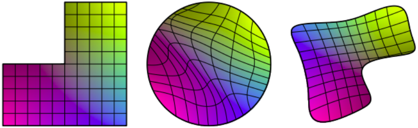
When $\theta_t \to 0$, the sequence $(\mathbb{P}_{\theta_t})_{t \in \mathbb{N}}$ converges to $\mathbb{P}_0$ under the EM distance, but does not converge at all under either the JS, KL, reverse KL, or TV divergences.

Plan

1. A brief history of GANs.
2. StyleGAN and the generative state-of-the-art.
3. Inversion, reconstruction and current work.

First change in StyleGAN.

Intuition: a $\mathcal{N}_n$ distribution is likely to be totally inappropriate for real datasets.



(a) Distribution of features in training set

(b) Mapping from $\mathcal{Z}$ to features

(c) Mapping from $\mathcal{W}$ to features

Figure 6. Illustrative example with two factors of variation (image features, e.g., masculinity and hair length). (a) An example training set where some combination (e.g., long haired males) is missing. (b) This forces the mapping from $\mathcal{Z}$ to image features to become curved so that the forbidden combination disappears in $\mathcal{Z}$ to prevent the sampling of invalid combinations. (c) The learned mapping from $\mathcal{Z}$ to $\mathcal{W}$ is able to "undo" much of the warping.

- Use another (!) neural network network f to 'disentangle' $\mathcal{Z}$ to $\mathcal{W}$.

UNIVERSITY OF CAMBRIDGE

Full StyleGAN Architecture.

Karras et al. (2017, 2018, 2019, 2020)[1] have drastically empirically improved the samples that GANs are able to generate. StyleGAN is essentially the concatenation of two neural networks:

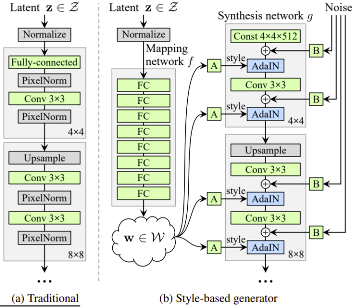
- Initial latent mapping network $f : \mathcal{Z} \to \mathcal{W}$.

[1] ALL important papers!

Full StyleGAN Architecture.

Karras et al. (2017, 2018, 2019, 2020)[1] have drastically empirically improved the samples that GANs are able to generate. StyleGAN is essentially the concatenation of two neural networks:

- Initial latent mapping network $f : \mathcal{Z} \to \mathcal{W}$.
- Synthesis network $h : \mathcal{W} \to \mathcal{X}$, where $\mathcal{X}$ the space of images.

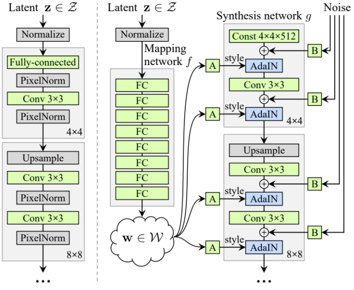


[1] ALL important papers!

Full StyleGAN Architecture.

Karras et al. (2017, 2018, 2019, 2020)[1] have drastically empirically improved the samples that GANs are able to generate. StyleGAN is essentially the concatenation of two neural networks:

- Initial latent mapping network $f : \mathcal{Z} \to \mathcal{W}$.
- Synthesis network $h : \mathcal{W} \to \mathcal{X}$, where $\mathcal{X}$ the space of images.
    - Additional choice to map $w \in \mathcal{W}$ **repeatedly** into the synthesis network (with additional noise) was also a significant contribution of the work.



(a) Traditional (b) Style-based generator

UNIVERSITY OF CAMBRIDGE

[1] ALL important papers!

Empirical results of style architectures.

The most well-known application of StyleGAN2 is the site `thispersondoesnotexist.com`:

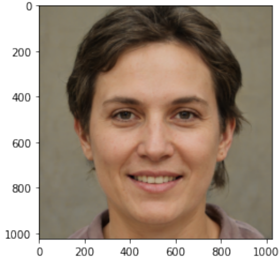


Figure: Sample of a face close to the 'average' face in the StyleGAN prior.

Figure 6. Progressive growing leads to "phase" artifacts. In this example the teeth do not follow the pose but stay aligned to the camera, as indicated by the blue line.

We can do even better!

Plan

1. A brief history of GANs.
2. StyleGAN and the generative state-of-the-art.
3. Inversion, reconstruction and current work.

The general GAN inversion problem.

Reference: GAN Inversion: A Survey (2021)

- Archetype: given ground truth x, solve

$$z^* = \text{argmin}_{z \in P}[\ell(G(z), x) + R(z)] \tag{5}$$

UNIVERSITY OF CAMBRIDGE

The general GAN inversion problem.

Reference: GAN Inversion: A Survey (2021)

- Archetype: given ground truth x, solve

$$z^* = \text{argmin}_{z \in P}[\ell(G(z), x) + R(z)] \tag{5}$$

- Choices:

The general GAN inversion problem.

Reference: GAN Inversion: A Survey (2021)

- Archetype: given ground truth x, solve

$$z^* = \text{argmin}_{z \in P}[\ell(G(z), x) + R(z)] \tag{5}$$

- Choices:

UNIVERSITY OF CAMBRIDGE

The general GAN inversion problem.

Reference: GAN Inversion: A Survey (2021)

- Archetype: given ground truth x, solve

$$z^* = \text{argmin}_{z \in P}[\ell(G(z), x) + R(z)] \tag{5}$$

- Choices:
-
  - General approach: optimisation or encoding (or both)?

The general GAN inversion problem.

Reference: GAN Inversion: A Survey (2021)

- Archetype: given ground truth x, solve

$$z^* = \text{argmin}_{z \in P}[\ell(G(z), x) + R(z)] \tag{5}$$

- Choices:
-
  - General approach: optimisation or encoding (or both)?
  - Loss function ℓ: pixelwise loss turns out to lead to very blurry images, even after regularization. Use VGG loss.

UNIVERSITY OF CAMBRIDGE

The general GAN inversion problem.

Reference: GAN Inversion: A Survey (2021)

- Archetype: given ground truth x, solve

$$z^* = \text{argmin}_{z \in P}[\ell(G(z), x) + R(z)] \tag{5}$$

- Choices:
-
  - General approach: optimisation or encoding (or both)?
  - Loss function ℓ: pixelwise loss turns out to lead to very blurry images, even after regularization. Use VGG loss.
  - *Which* latent space P?

UNIVERSITY OF CAMBRIDGE

The general GAN inversion problem.

Reference: GAN Inversion: A Survey (2021)

- Archetype: given ground truth x, solve

$$z^* = \text{argmin}_{z \in P}[\ell(G(z), x) + R(z)] \tag{5}$$

- Choices:
-
    - General approach: optimisation or encoding (or both)?
    - Loss function ℓ: pixelwise loss turns out to lead to very blurry images, even after regularization. Use VGG loss.
    - *Which* latent space P?
    - How to regularize?

UNIVERSITY OF CAMBRIDGE

An example from my training.

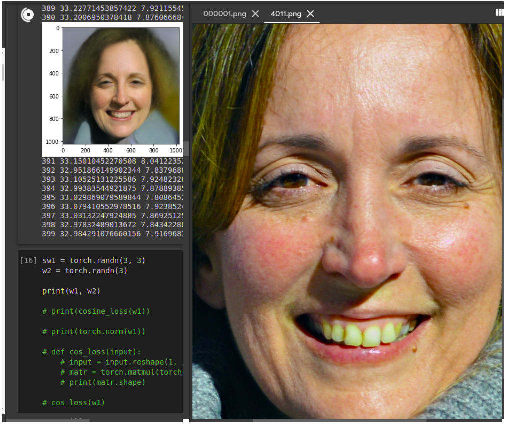


Figure: Inversion in less than 10 minutes (using almost only VGG loss).

UNIVERSITY OF CAMBRIDGE

The SOTA for inpainting.

Reference: R. Marinescu, D. Moyer, P. Golland [2020]

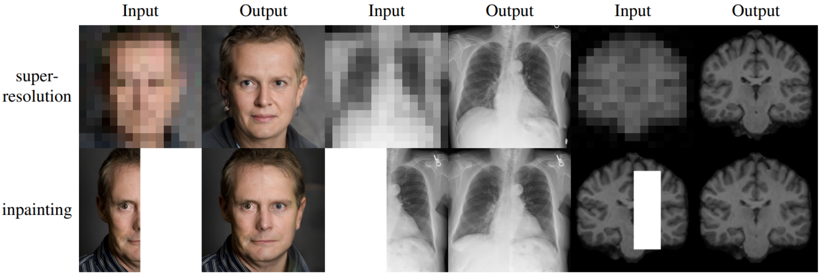
- 

Figure: The inpainting capabilities of inverting StyleGAN.

The SOTA for inpainting.

Reference: R. Marinescu, D. Moyer, P. Golland [2020]

$$
\begin{aligned}
w^* &= \arg\max_w p(w)p(I|w) \\
&= \prod_i \mathcal{N}(w_i|\mu,\sigma^2) \prod_{i,j} \mathcal{M}(cos^{-1}\frac{w_i w_j^T}{|w_i||w_j|}|0,\kappa) \\
&\quad \mathcal{N}(I|f \circ G(w), \sigma^2_{pixel}\mathbb{I}_{n_f^2}) \\
&\quad \mathcal{N}(\phi(I)|\phi \circ f \circ G(w), \sigma^2_{percept}\mathbb{I}_{n_\phi^2})
\end{aligned}
$$

Figure: Regularized, efficient optimization?

UNIVERSITY OF CAMBRIDGE

The SOTA for inpainting.

Reference: R. Marinescu, D. Moyer, P. Golland [2020]

$$w^* = \arg\min_w \underbrace{\sum_i \left(\frac{w_i - \mu}{\sigma_i}\right)^2}_{\mathcal{L}_w} \underbrace{-2\kappa \sum_{i,j} \frac{w_i w_j^T}{|w_i||w_j|}}_{\mathcal{L}_{colin}}$$
$$+ \sigma_{pixel}^{-2} \underbrace{\|I - f \circ G(w)\|_2^2}_{\mathcal{L}_{pixel}} \quad (8)$$
$$+ \sigma_{percept}^{-2} \underbrace{\|I - \phi \circ f \circ G(w)\|_2^2}_{\mathcal{L}_{percept}}$$

which can be succintly written as a weighted sum of four loss terms:

$$w^* = \arg\min_w \mathcal{L}_w + \lambda_c \mathcal{L}_{colin} + \lambda_x \mathcal{L}_{pixel} + \lambda_p \mathcal{L}_{percept} \quad (9)$$

where $\mathcal{L}_w$ is the prior loss over w, $\mathcal{L}_{colin}$ is the colinearity loss on w, $\mathcal{L}_{pixel}$ is the pixelwise loss on the image reconstruction, and $\mathcal{L}_{percept}$ is the perceptual loss, $\lambda_c = -2\kappa$, $\lambda_{pixel} = \sigma_{pixel}^{-2}$ and $\lambda_{percept} = \sigma_{percept}^{-2}$.

Figure: Resultant loss.

What makes this work?

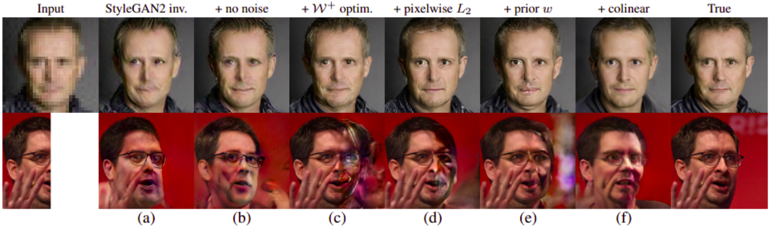


Figure 4: Reconstructions as the loss function evolves from the original StyleGAN2 inversion to our proposed method. Top row shows super resolution, while bottom row shows inpainting. We start from (a) the original StyleGAN2 inversion, and (b) remove noise optimisation, (c) extend optimisation to full $\mathcal{W}^+$ space, (d) add pixelwise L_2 term, (e) add prior on w latent variables and (f) add colinear loss term for w.

Figure: Illustration of uncurated results for approaching the problem.

UNIVERSITY OF CAMBRIDGE

What next?

Can we do better than the fairly naive approach to regularizing w?

- Perceptual path length?

What next?

Can we do better than the fairly naive approach to regularizing w?

- Perceptual path length?
- D as a regularizer? Probably not ...

What next?

Can we do better than the fairly naive approach to regularizing w?

- Perceptual path length?
- D as a regularizer? Probably not ...
- Still, a learning-based approach may exist.

Can we do better than the fairly naive approach to regularizing w?

- Perceptual path length?
- D as a regularizer? Probably not ...
- Still, a learning-based approach may exist.

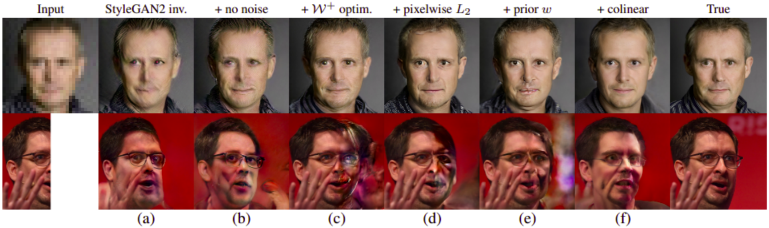


Figure 4: Reconstructions as the loss function evolves from the original StyleGAN2 inversion to our proposed method. Top row shows super resolution, while bottom row shows inpainting. We start from (a) the original StyleGAN2 inversion, and (b) remove noise optimisation, (c) extend optimisation to full $\mathcal{W}^+$ space, (d) add pixelwise L_2 term, (e) add prior on w latent variables and (f) add colinear loss term for w.

-

Figure: Variety of techniques applied.

Thanks!

- Thanks to Dr Mukherjee, Dr Aviles-Rivero and Professor Schönlieb.

Thanks!

- Thanks to Dr Mukherjee, Dr Aviles-Rivero and Professor Schönlieb.
- Slides hopefully at `https://arthurconmy.github.io/`.