Wasserstein-2 generative networks

Evgeny Burnaev

Skoltech
Head of ADASE Group

joint with Alexander Korotin, Vage Egiazaryan

Outline

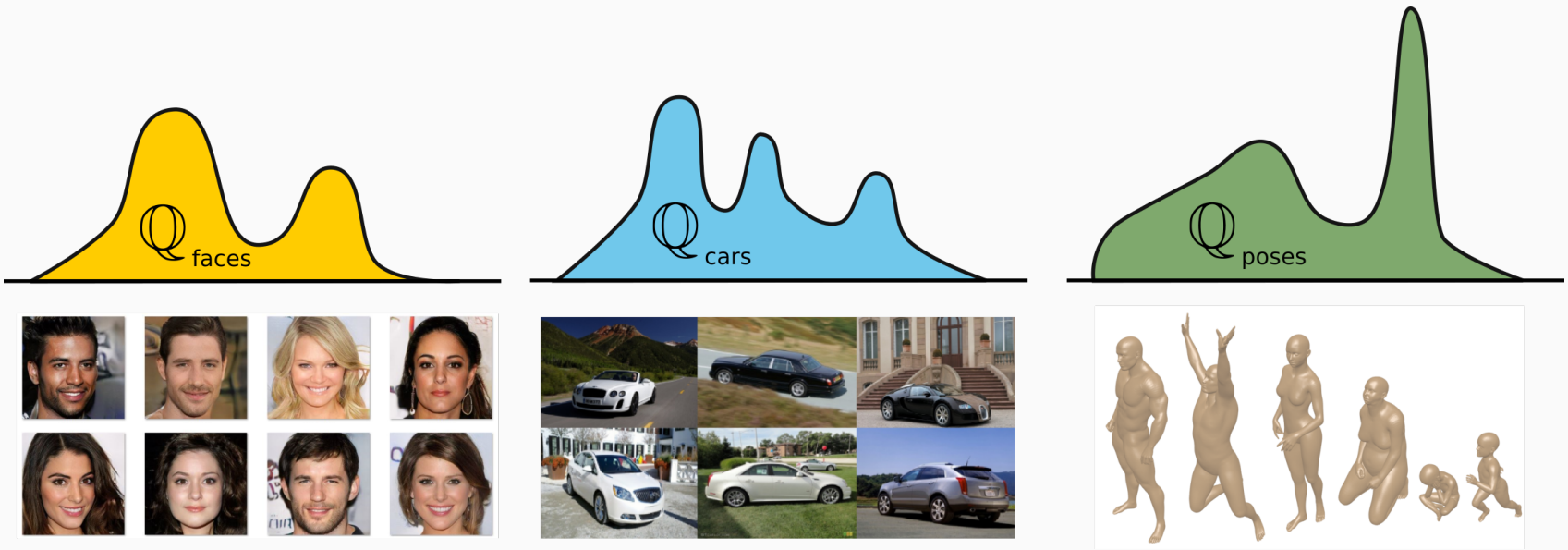
Generative Learning

Generative Networks

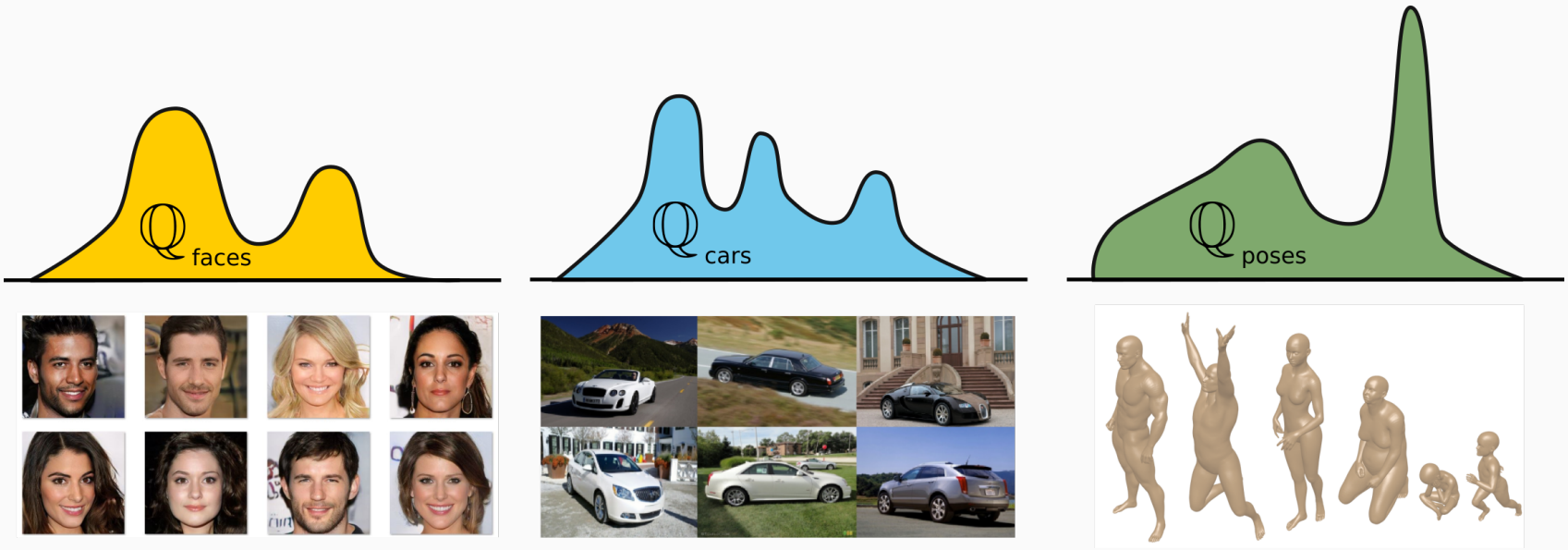
Wasserstein-2 Generative Networks

Generative Learning

Unsupervised Learning: Distribution Estimation



Unsupervised Learning: Distribution Estimation



General problem

Reconstruct the distribution $\mathbb{Q}$ by using observed samples

$$x_1, \ldots, x_N$$

Explicit Distribution Estimation

Compute/approximate the density

Explicit Distribution Estimation

Compute/approximate the density

1. **Parametric models** (GMM, NADE, RBM)
   Pick the density function from parametric class $\{p_\theta \mid \theta \in \Theta\}$:

$$\theta^* = \arg\max_{\theta \in \Theta} \log p(\theta | x_1, \ldots, x_N)$$

Explicit Distribution Estimation

Compute/approximate the density

1. **Parametric models** (GMM, NADE, RBM)
   Pick the density function from parametric class $\{p_\theta \mid \theta \in \Theta\}$:

$$\theta^* = \arg\max_{\theta \in \Theta} \log p(\theta | x_1, \ldots, x_N)$$

2. **Non-parametric models** (Histograms, KDE)
   The complexity of the model depends on the sample size.

Explicit Distribution Estimation

Compute/approximate the density

1. **Parametric models** (GMM, NADE, RBM)
   Pick the density function from parametric class $\{p_\theta \mid \theta \in \Theta\}$:

$$\theta^* = \arg\max_{\theta \in \Theta} \log p(\theta | x_1, \ldots, x_N)$$

2. **Non-parametric models** (Histograms, KDE)
   The complexity of the model depends on the sample size.

Problems

- Not existing/vanishing density
- Too complex density for parametric approaches
- Too computationally hard non-parametric approximations

Implicit Distribution Estimation

Define a stochastic procedure to **generate** data!

Implicit Distribution Estimation

Define a stochastic procedure to **generate** data!

- Normalizing Flows
- Generative Stochastic Networks
- Variational Autoencoders
- **Generative Adversarial Networks**

Implicit Distribution Estimation

Define a stochastic procedure to **generate** data!

- Normalizing Flows
- Generative Stochastic Networks
- Variational Autoencoders
- **Generative Adversarial Networks**

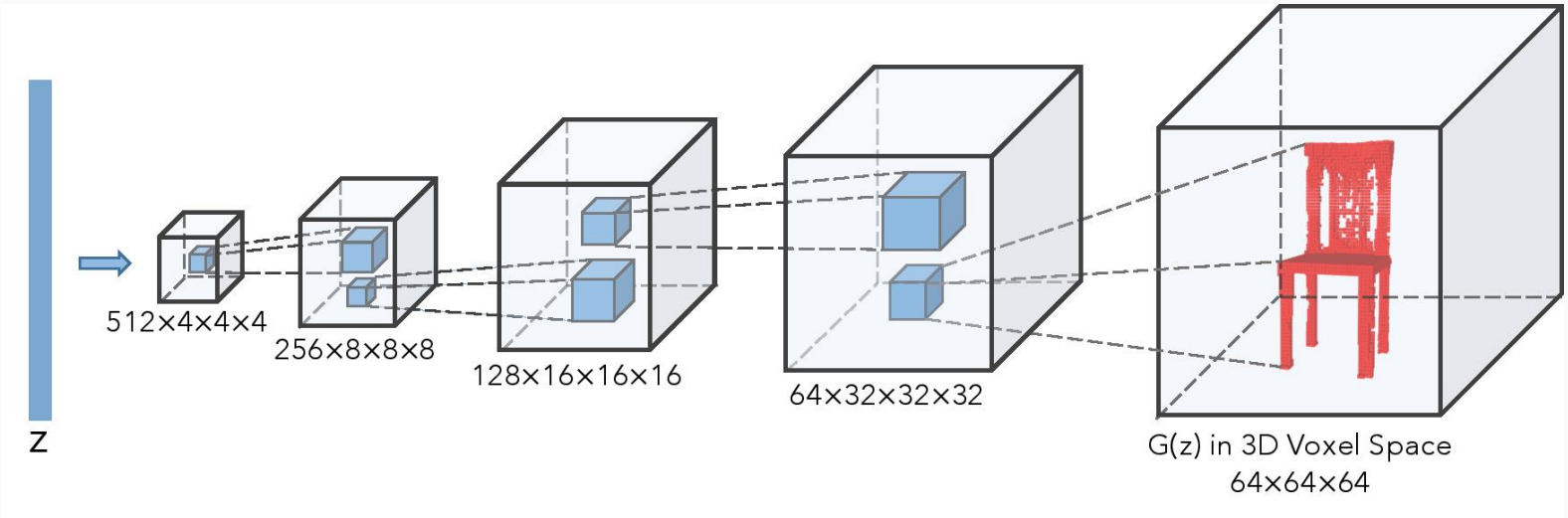
A schematic example of a generative approach:

Implicit Distribution Estimation

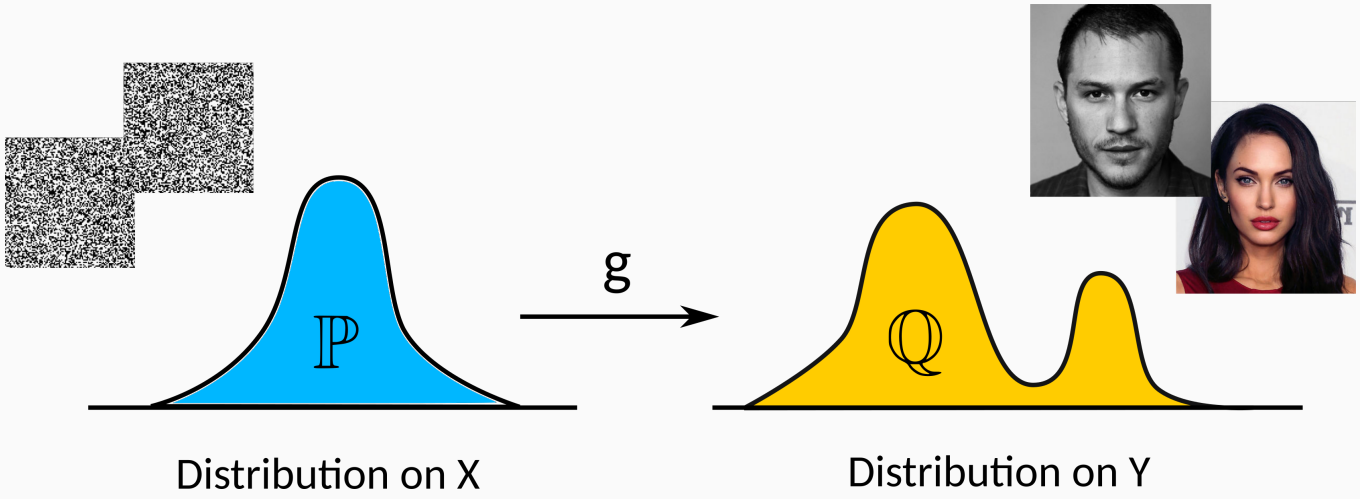
Define a stochastic procedure to **generate** data!

- Normalizing Flows
- Generative Stochastic Networks
- Variational Autoencoders
- **Generative Adversarial Networks**

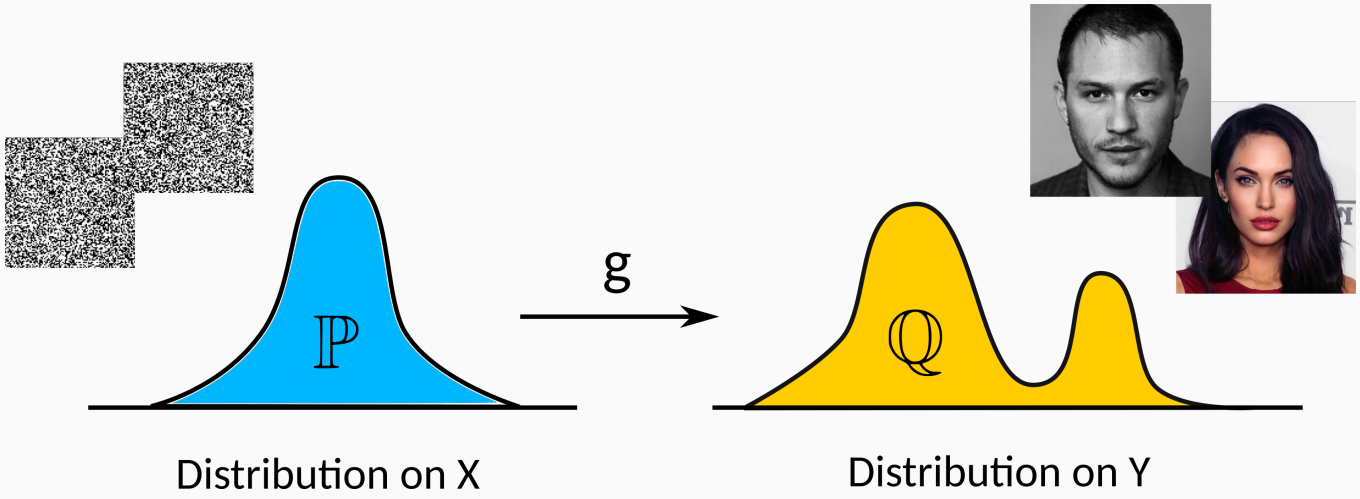
A schematic example of a generative approach:



Generative Approach



Generative Approach



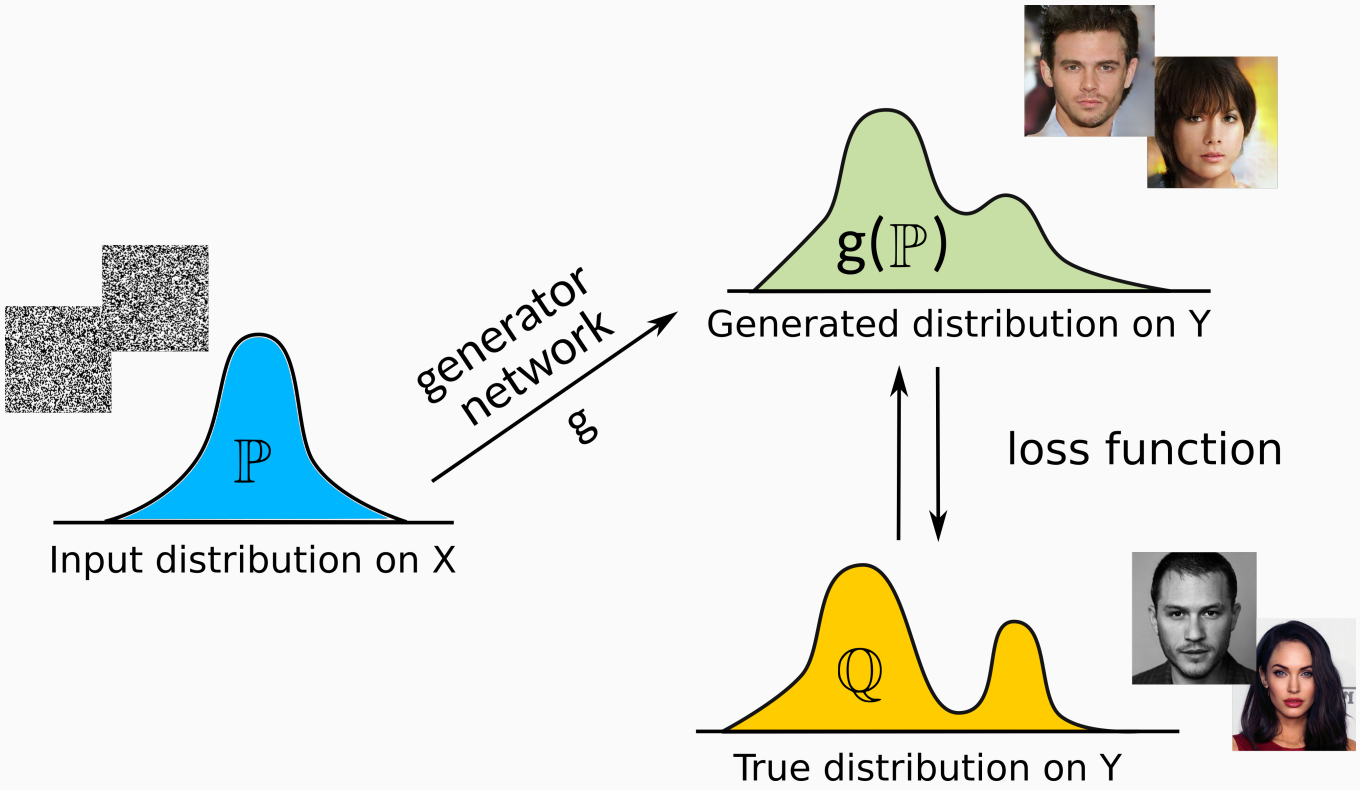
Generative mapping

$$g : \mathcal{X} \to \mathcal{Y} \qquad \text{with} \qquad g \circ \mathbb{P} = \mathbb{Q}$$

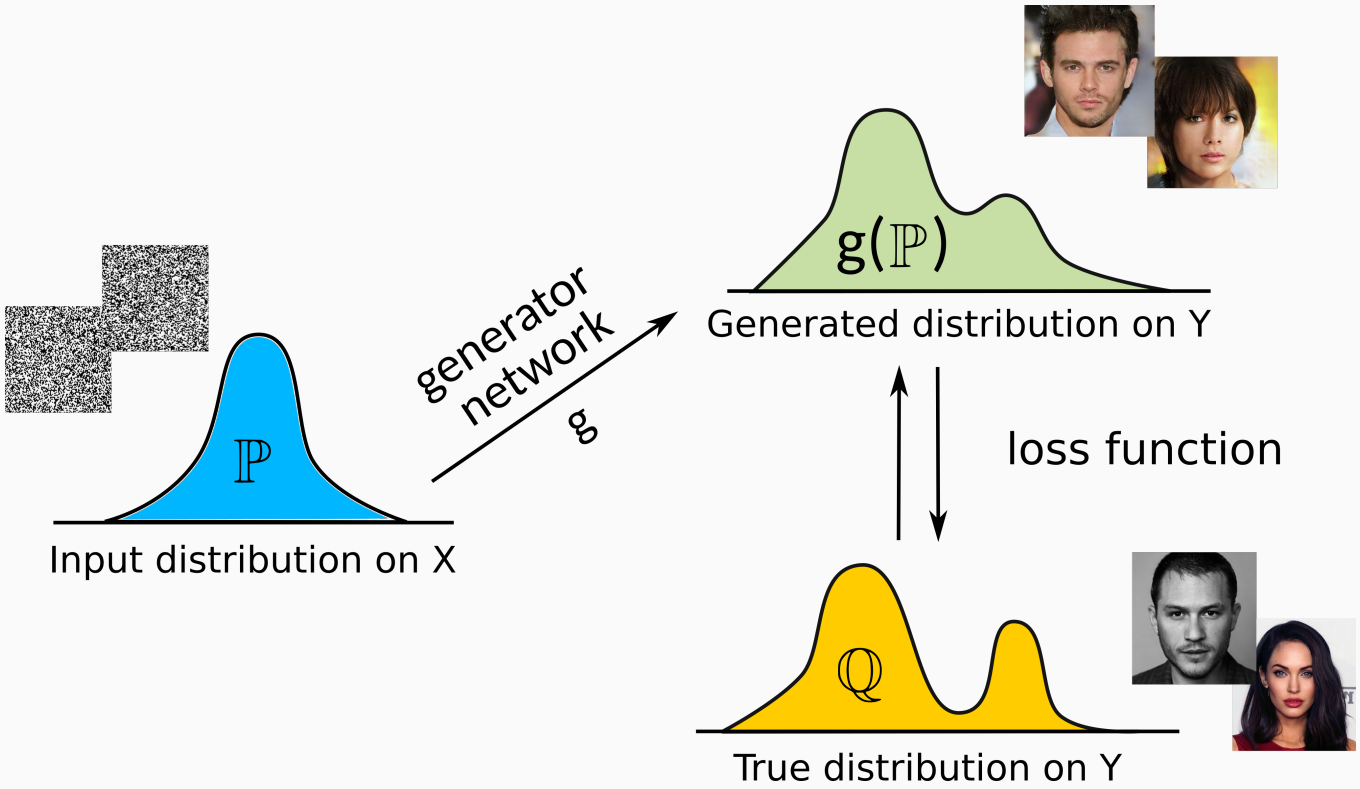
$$x \sim \mathbb{P} \Rightarrow g(x) \sim \mathbb{Q}$$

Generative Networks

Generative Networks



Generative Networks

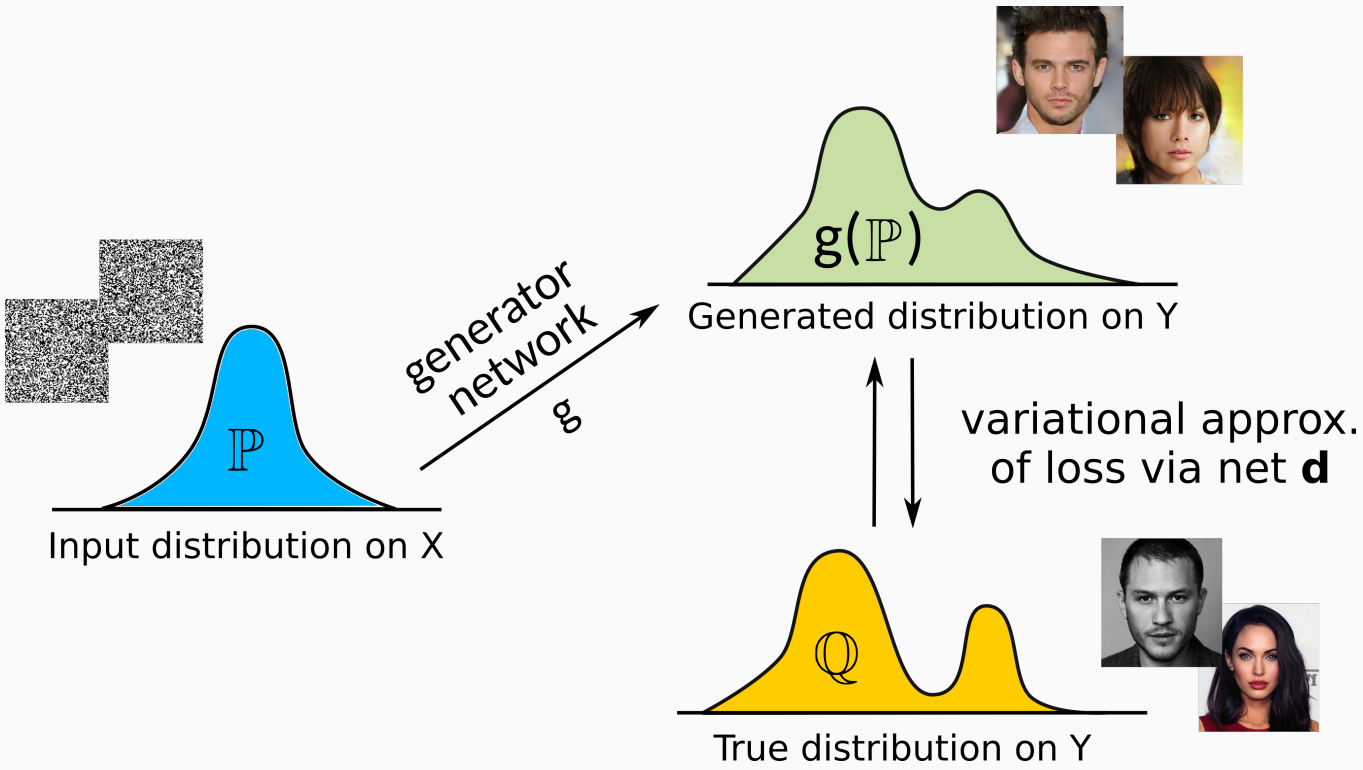


Optimization Objective

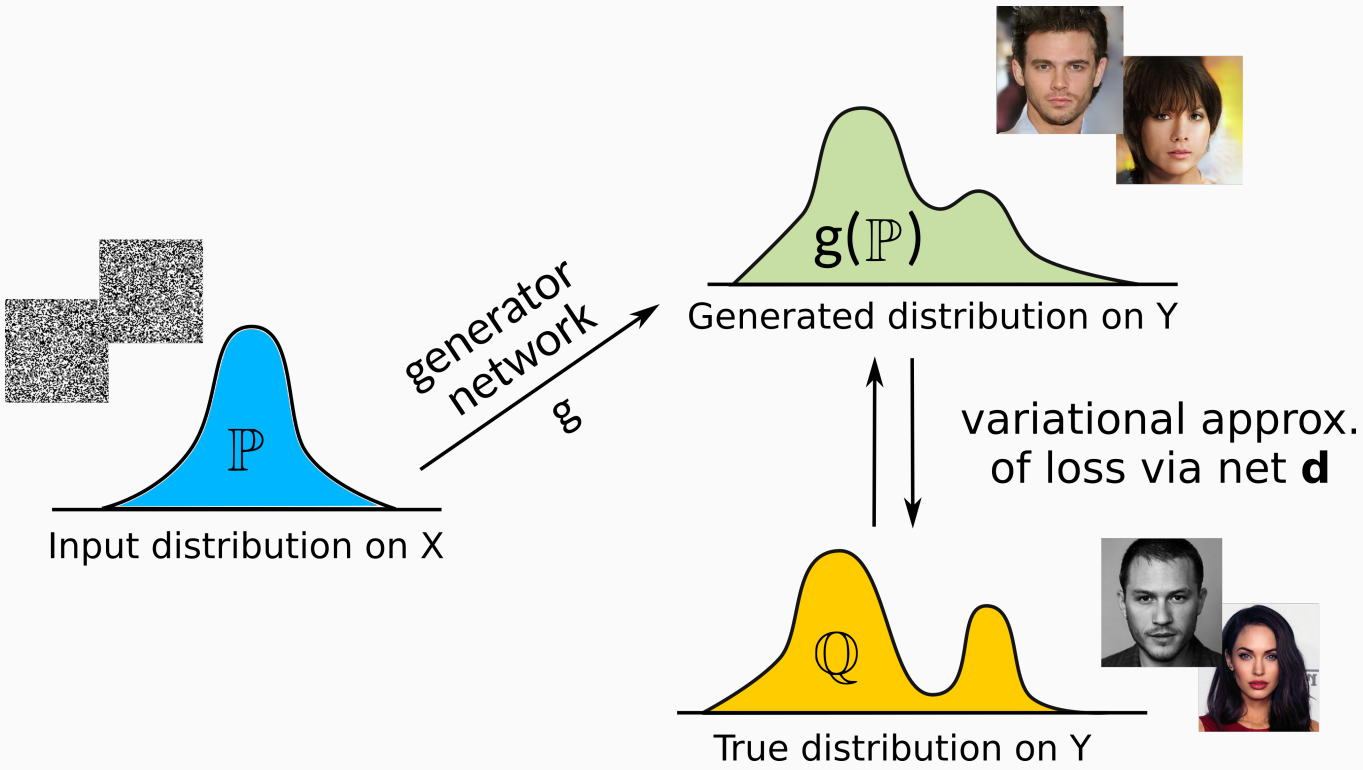
$$\underset{g \in \mathcal{G}}{\text{minimize}}\, \textbf{loss}\big(g(\mathbb{P}), \mathbb{Q}\big)$$

loss $=$ f-divergence (KL, JS, etc.), Wasserstein distance, etc.

Generative Adversarial Networks



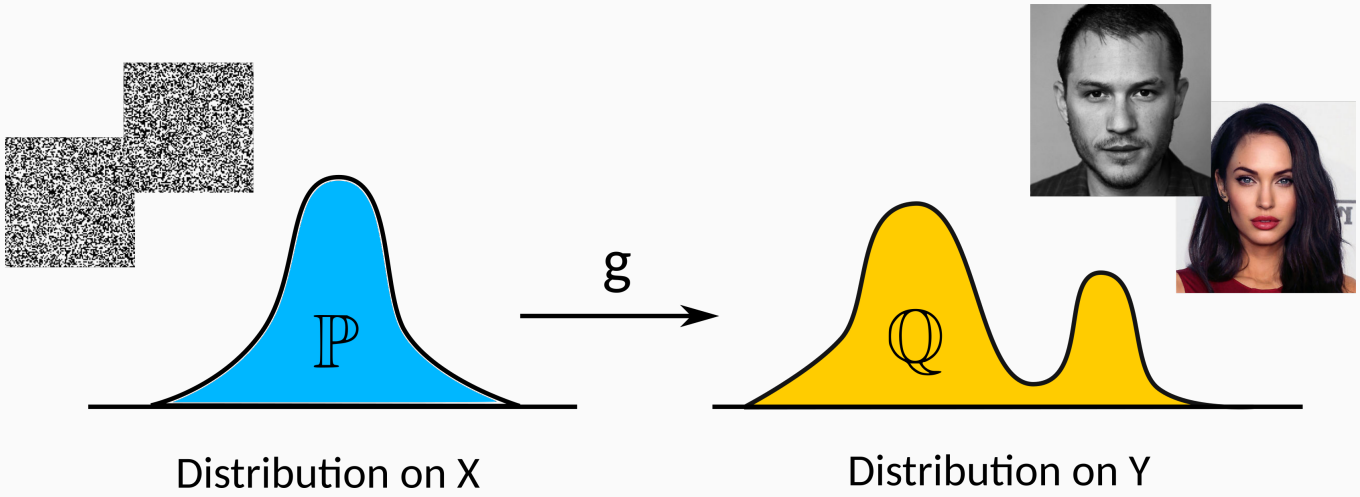
Generative Adversarial Networks



Minimax Optimization Objective

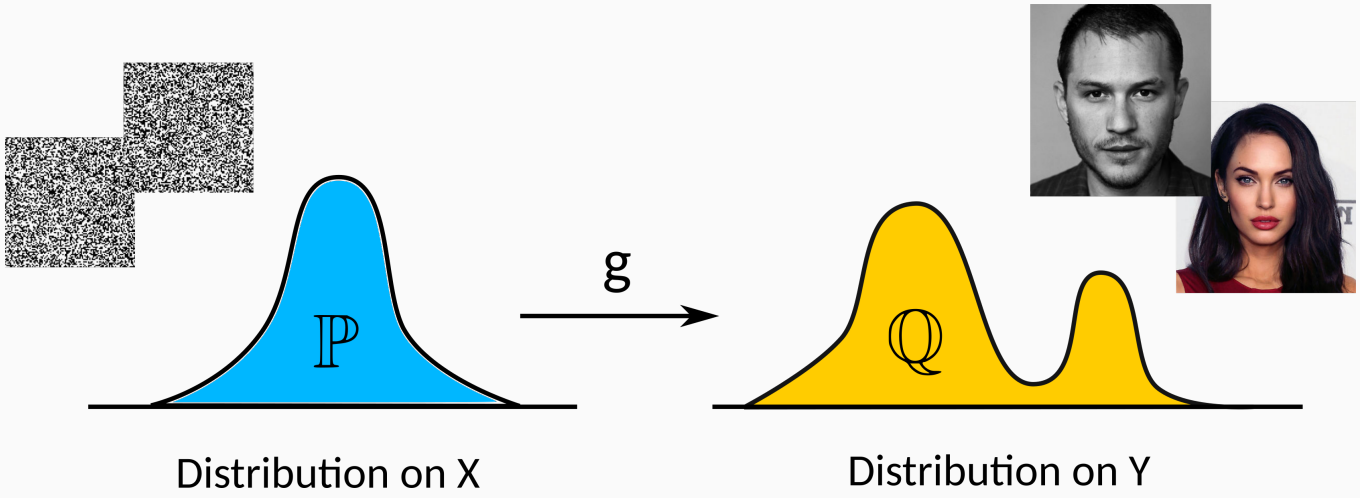
$$\underset{g\in\mathcal{G}}{\text{minimize}}\,\textbf{loss}\big(g(\mathbb{P}),\mathbb{Q}\big) = \underset{g\in\mathcal{G}}{\text{minimize}}\,\max_{d\in\mathcal{D}} V_{\mathbb{P},\mathbb{Q}}(g,d)$$

Successes in Image Generation[1]



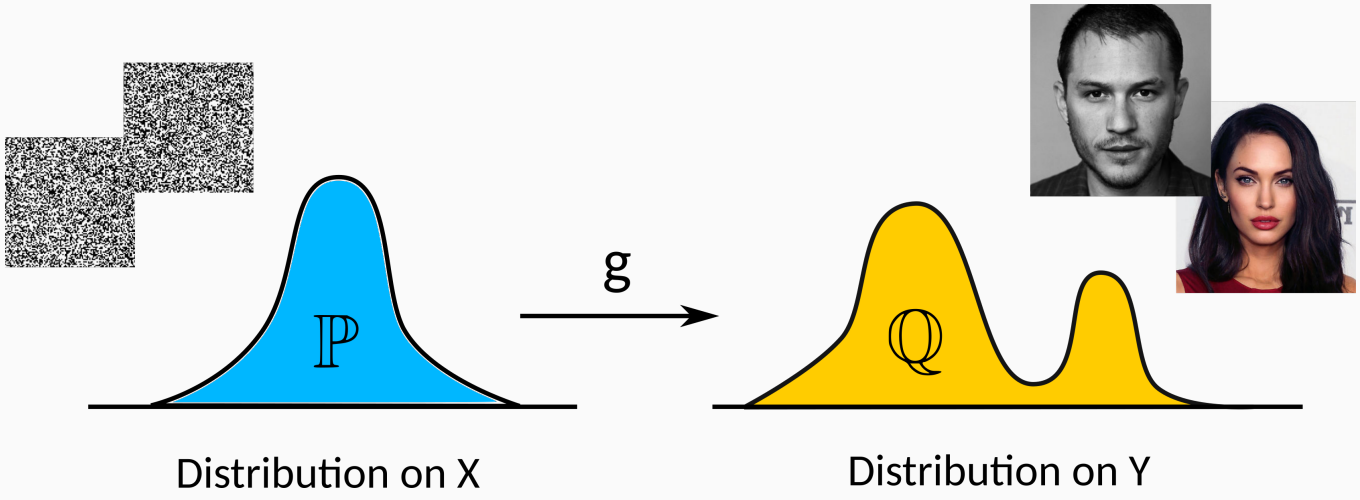
[1] Andrew Brock, Jeff Donahue, and Karen Simonyan (2018). "Large scale gan training for high fidelity natural image synthesis". In: *arXiv preprint arXiv:1809.11096*.

Successes in Image Generation[1]



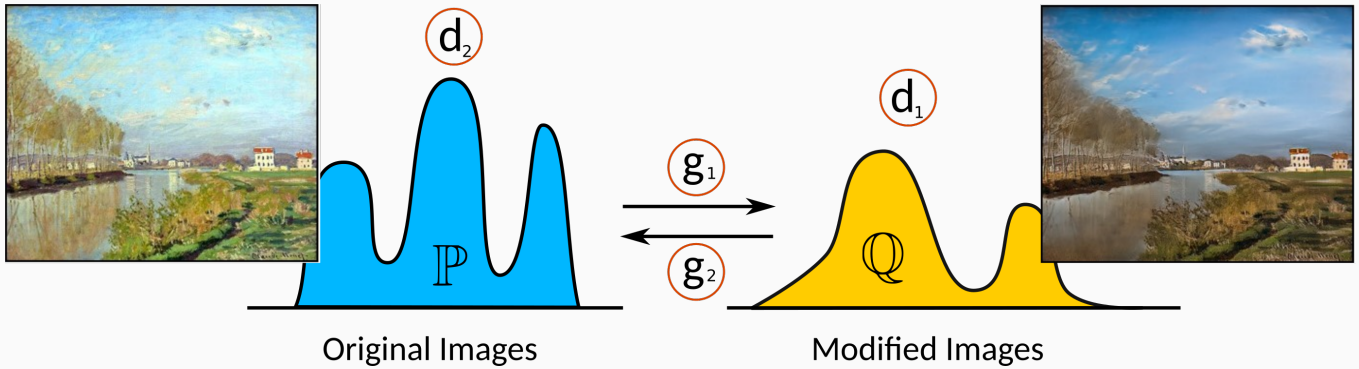
[1] Andrew Brock, Jeff Donahue, and Karen Simonyan (2018). "Large scale gan training for high fidelity natural image synthesis". In: *arXiv preprint arXiv:1809.11096*.

Successes in Image Generation[1]



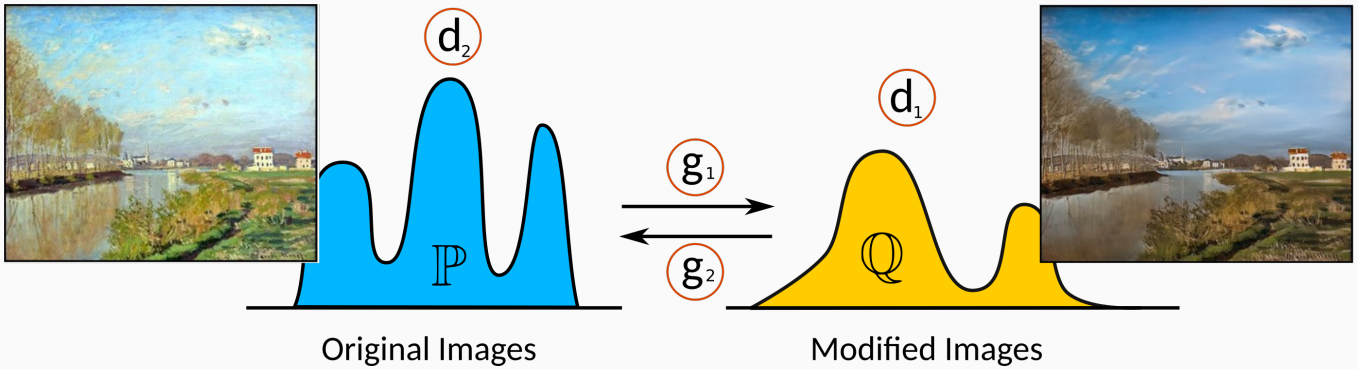
[1]Andrew Brock, Jeff Donahue, and Karen Simonyan (2018). "Large scale gan training for high fidelity natural image synthesis". In: *arXiv preprint arXiv:1809.11096*.

Successes in Image-to-Image Style Transfer (Cycle GAN[2])



[2] Jun-Yan Zhu et al. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

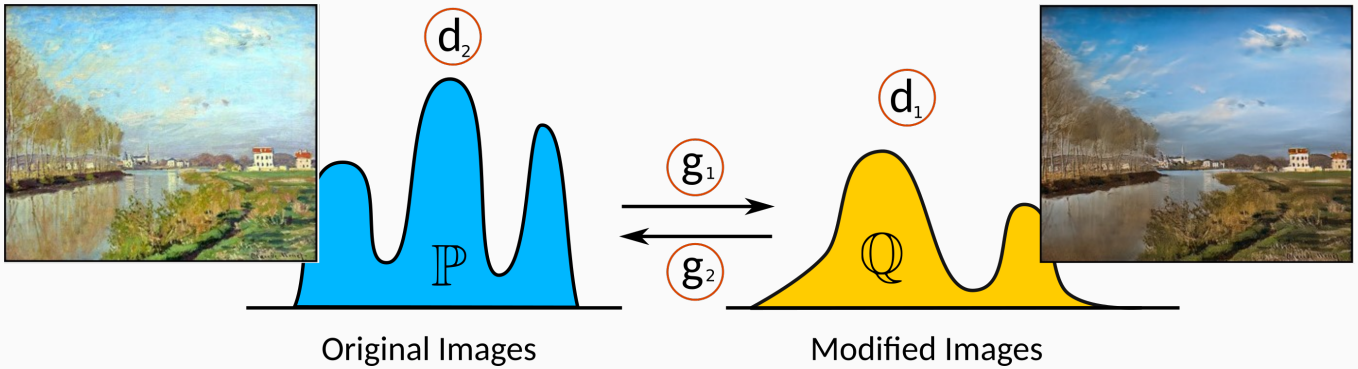
Successes in Image-to-Image Style Transfer (Cycle GAN[2])



Winter to Summer

[2] Jun-Yan Zhu et al. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

Successes in Image-to-Image Style Transfer (Cycle GAN[2])



Winter to Summer

Sunny to Rainy

[2] Jun-Yan Zhu et al. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

Generative Adversarial Networks: Problems

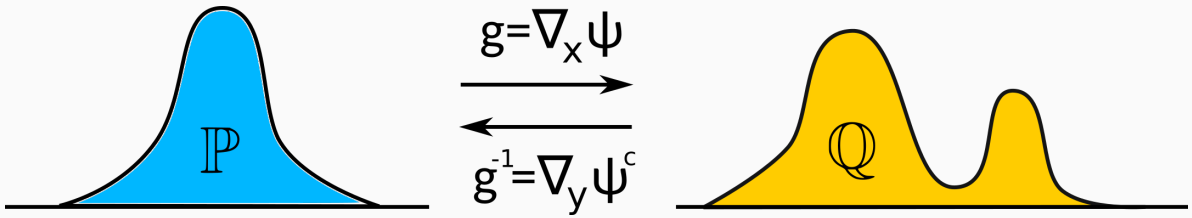
- **Hardcore Training**
  - Non convex-concave optimization
  - Mode dropping
  - Model selection
  - ...
- **Inference & application**
  - Poor Generalization
  - Non-invertibility
  - Poor mapping/latent structure
  - ...

GAN Fails

(generated images)

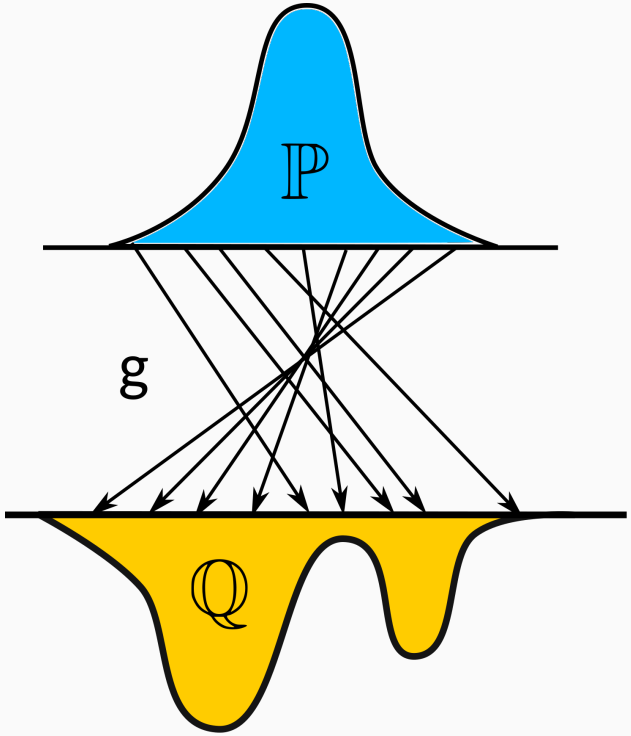
Wasserstein-2 Generative Networks

Wasserstein-2 Generative Networks

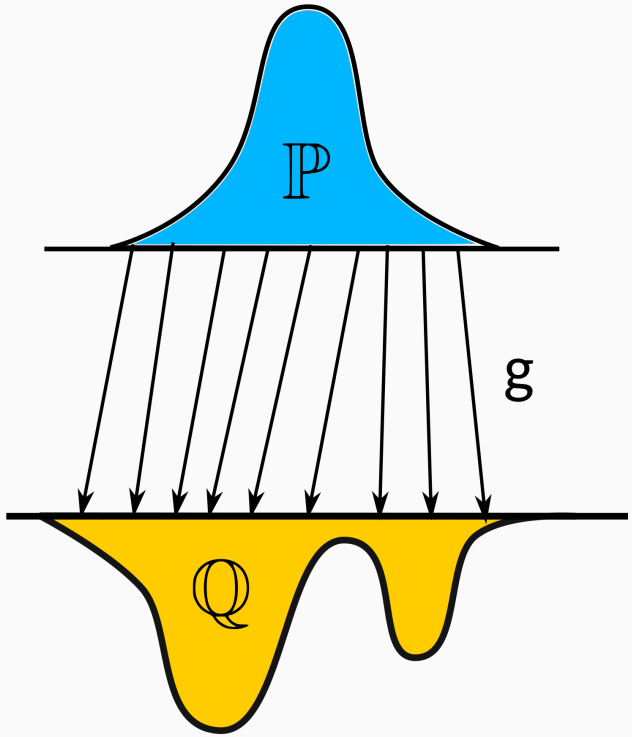


A. Korotin, V. Egiazarian, A. Asadulaev, E. Burnaev (2019). "Wasserstein-2 Generative Networks". In: arXiv preprint arXiv:1909.13082

Optimal Mappings?

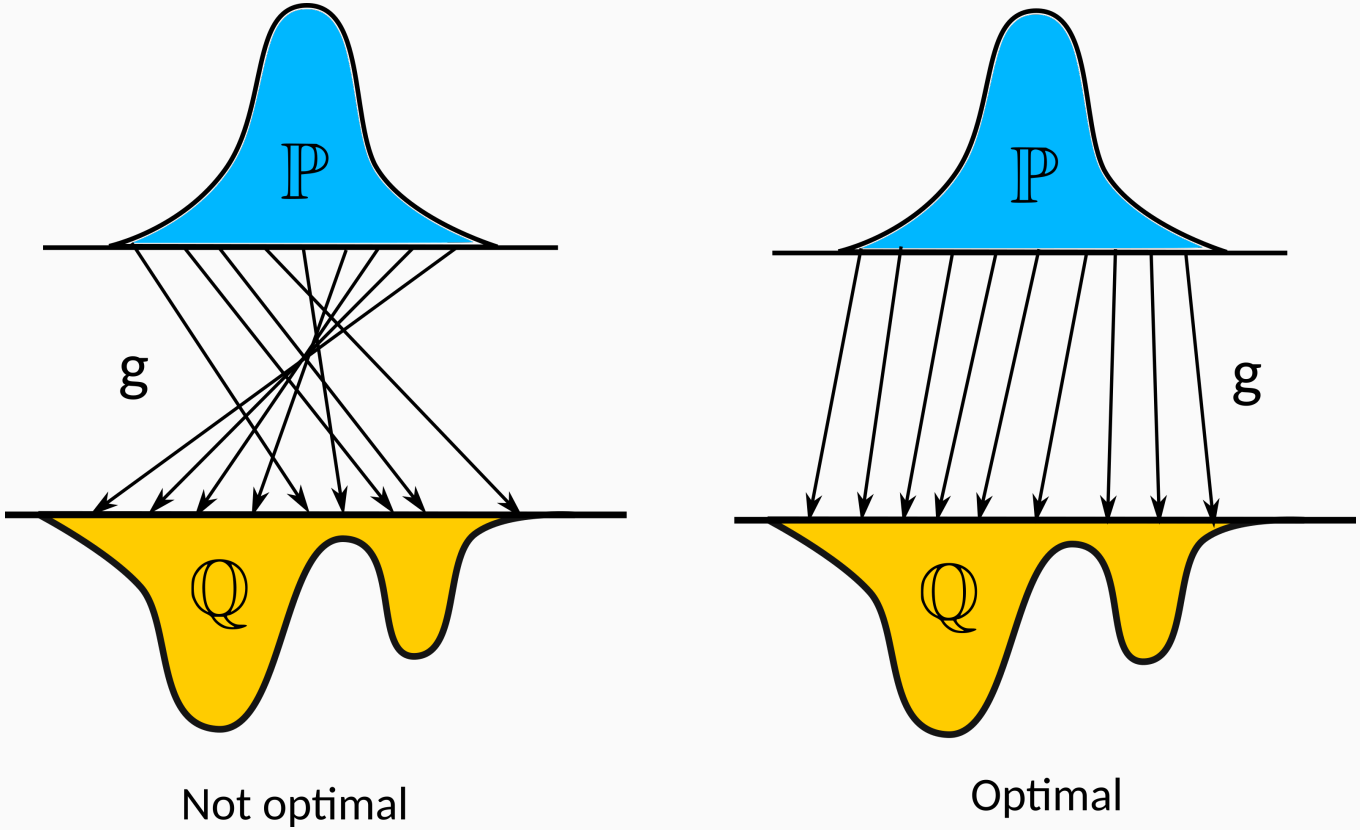


Not optimal



Optimal

Optimal Mappings?



Not optimal

Optimal

What is **optimal** generative mapping?

Optimal = Simple, Intuitive, Not Overparametrized

Cyclically Monotone Mapping

Assume $\mathcal{X} = \mathcal{Y} = \mathbb{R}^D$. For all $x, x' \in \mathcal{X}$ require strict

[3] Cédric Villani (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

Cyclically Monotone Mapping

Assume $\mathcal{X} = \mathcal{Y} = \mathbb{R}^D$. For all $x, x' \in \mathcal{X}$ require strict

- **Monotonicity** for $D = 1$:

$$\big(g(x) - g(x')\big) \cdot \big(x - x'\big) > 0;$$

[3] Cédric Villani (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

Cyclically Monotone Mapping

Assume $\mathcal{X} = \mathcal{Y} = \mathbb{R}^D$. For all $x, x' \in \mathcal{X}$ require strict

- **Monotonicity** for $D = 1$:

$$\big(g(x) - g(x')\big) \cdot \big(x - x'\big) > 0;$$

- **Cycle Monotonicity**[3] for $D \geq 1$:

$$\langle g(x) - g(x'), x - x' \rangle > 0.$$

[3] Cédric Villani (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

Cyclically Monotone Mapping

Assume $\mathcal{X} = \mathcal{Y} = \mathbb{R}^D$. For all $x, x' \in \mathcal{X}$ require strict

- **Monotonicity** for $D = 1$:

$$\big(g(x) - g(x')\big) \cdot \big(x - x'\big) > 0;$$

- **Cycle Monotonicity**[3] for $D \geq 1$:

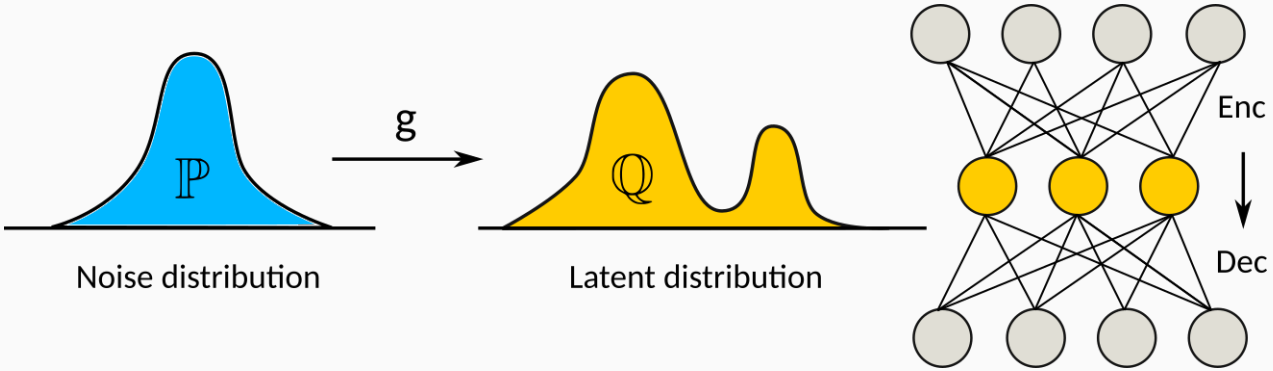
$$\langle g(x) - g(x'), x - x' \rangle > 0.$$

Well-structured and **invertible** generative mapping g!

[3] Cédric Villani (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

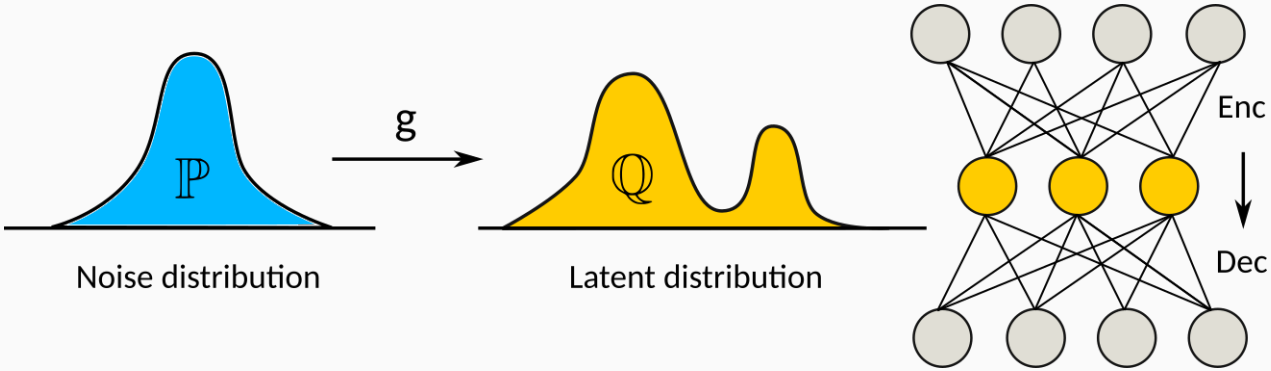
When $\mathcal{X} = \mathcal{Y}$?

- Latent Space Generative Mapping

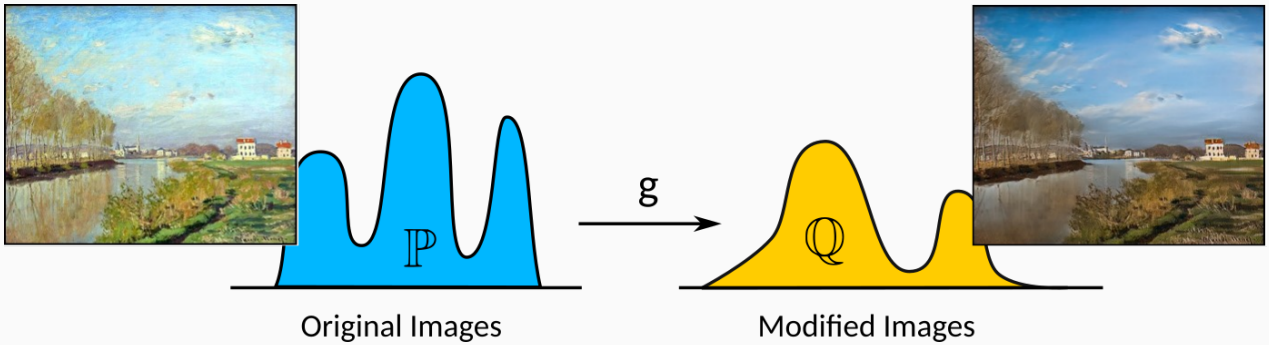


When $\mathcal{X} = \mathcal{Y}$?

- Latent Space Generative Mapping



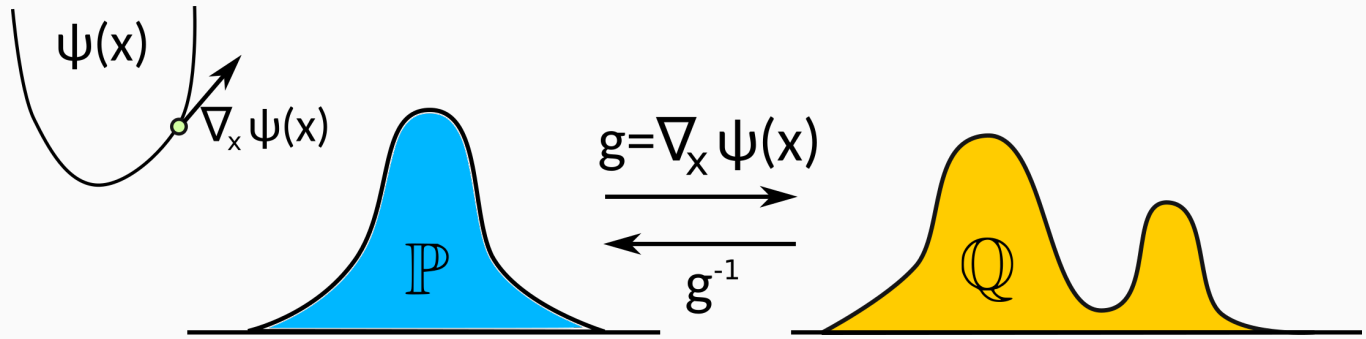
- Image-To-Image Translation



Existince and Uniqueness

Theorem[4]. For (almost) every $\mathbb{P}, \mathbb{Q}$ on $\mathcal{X} = \mathcal{Y} \subset \mathbb{R}^D$ cyclically monotone generator $g \circ \mathbb{P} = \mathbb{Q}$

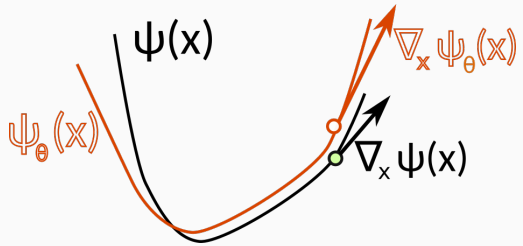
- Exists and is Unique!
- Is a gradient $g = \nabla_x \psi(x)$ of a convex function $\psi : \mathbb{R}^D \to \mathbb{R}$!



[4]Yann Brenier (1991). "Polar factorization and monotone rearrangement of vector-valued functions". In: *Communications on pure and applied mathematics* 44.4, pp. 375–417.

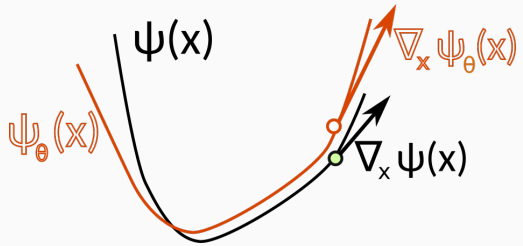
The Idea

Approximate convex function $\psi(x) : \mathbb{R}^D \to \mathbb{R}$ by neural nets!



The Idea

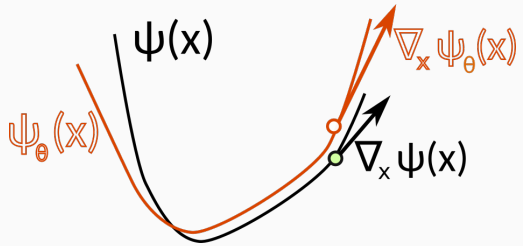
Approximate convex function $\psi(x) : \mathbb{R}^D \to \mathbb{R}$ by neural nets!



- $\psi_\theta : \mathbb{R}^D \to \mathbb{R}$ – deep input convex neural network (ICNN);

The Idea

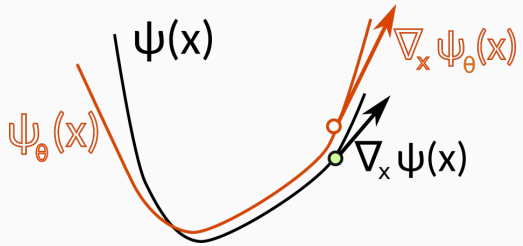
Approximate convex function $\psi(x) : \mathbb{R}^D \to \mathbb{R}$ by neural nets!



- $\psi_\theta : \mathbb{R}^D \to \mathbb{R}$ – deep input convex neural network (ICNN);
- $g_\theta = \nabla_x \psi_\theta : \mathbb{R}^D \to \mathbb{R}^D$ — generative mapping.

The Idea

Approximate convex function $\psi(x) : \mathbb{R}^D \to \mathbb{R}$ by neural nets!



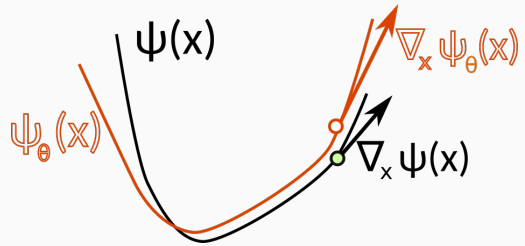
- $\psi_\theta : \mathbb{R}^D \to \mathbb{R}$ – deep input convex neural network (ICNN);
- $g_\theta = \nabla_x \psi_\theta : \mathbb{R}^D \to \mathbb{R}^D$ — generative mapping.

Questions

1. How to approximate convex functions by nets? - Next slide

The Idea

Approximate convex function $\psi(x) : \mathbb{R}^D \to \mathbb{R}$ by neural nets!



- $\psi_\theta : \mathbb{R}^D \to \mathbb{R}$ – deep input convex neural network (ICNN);
- $g_\theta = \nabla_x \psi_\theta : \mathbb{R}^D \to \mathbb{R}^D$ — generative mapping.

Questions

1. How to approximate convex functions by nets? - Next slide
2. How to find the generator? - GANs, Normalizing Flows, etc.

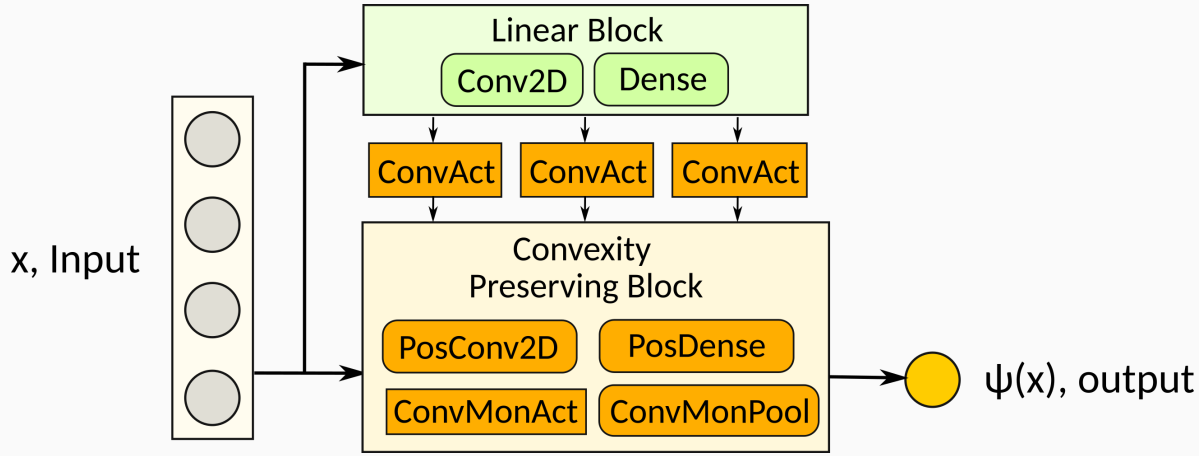
Input Convex Neural Networks

Based on the sequential variant[5] for ICNN $\psi_\theta(x) : \mathbb{R}^D \to \mathbb{R}$

[5] Brandon Amos, Lei Xu, and J Zico Kolter (2017). "Input convex neural networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 146–155.

Input Convex Neural Networks

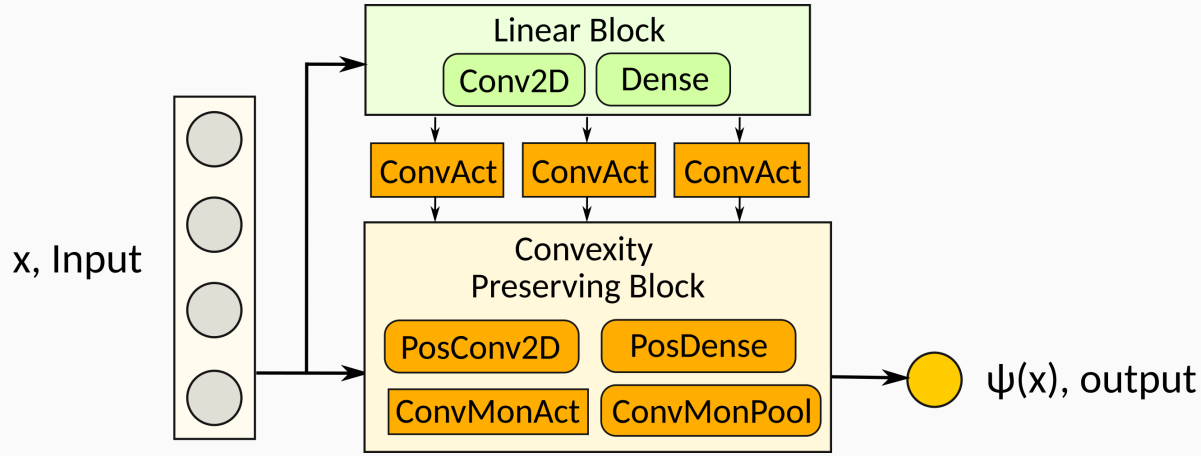
Based on the sequential variant[5] for ICNN $\psi_\theta(x) : \mathbb{R}^D \to \mathbb{R}$



[5] Brandon Amos, Lei Xu, and J Zico Kolter (2017). "Input convex neural networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 146–155.

Input Convex Neural Networks

Based on the sequential variant[5] for ICNN $\psi_\theta(x) : \mathbb{R}^D \to \mathbb{R}$



Inside CP Block: only positive weights (except biases) in linear layers and only convex & monotone activations (e.g. CELU)

[5] Brandon Amos, Lei Xu, and J Zico Kolter (2017). "Input convex neural networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 146–155.

Wasserstein-2 Distance

Monotone mapping

$$g^* = \underset{g \circ \mathbb{P} = \mathbb{Q}}{\operatorname{argmin}} \int_{\mathcal{X}} \frac{\|x - g(x)\|^2}{2} d\mathbb{P}(x)$$

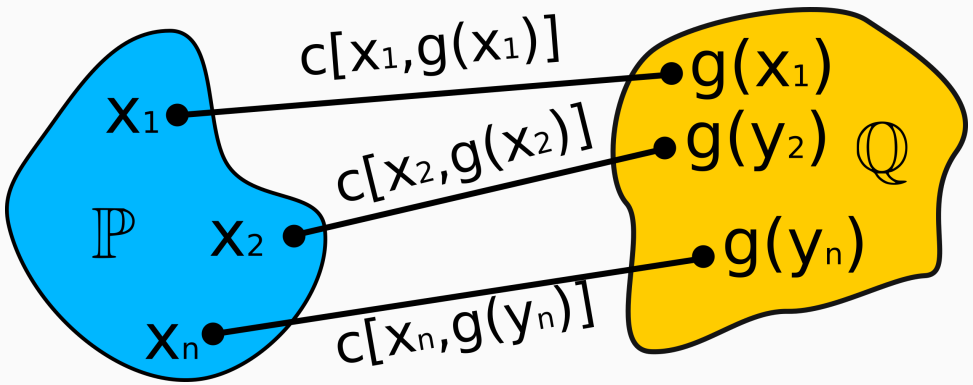
Wasserstein-2 Distance

Monotone mapping

$$g^* = \underset{g \circ \mathbb{P} = \mathbb{Q}}{\operatorname{argmin}} \int_{\mathcal{X}} \frac{\|x - g(x)\|^2}{2} d\mathbb{P}(x)$$

attains optimal quadratic transport cost (**Wasserstein-2 distance**)

$$W_2(\mathbb{P}, \mathbb{Q}) = \min_{g \circ \mathbb{P} = \mathbb{Q}} \int_{\mathcal{X}} \frac{\|x - g(x)\|^2}{2} d\mathbb{P}(x)$$



Dual Form of Wasserstein-2 distance

$$W_2(\mathbb{P},\mathbb{Q}) = -\min_{\psi\in \mathsf{Conv}} \Bigg[\underbrace{\int_{\mathcal{X}} \psi(x)d\mathbb{P}(x) + \int_{\mathcal{Y}} \psi^c(y)d\mathbb{Q}(y)}_{\mathrm{Corr}(\mathbb{P},\mathbb{Q})}\Bigg] + \mathsf{Const}(\mathbb{P},\mathbb{Q})$$

$$\psi^c(y) = \max_x \left(\langle x, y\rangle - \psi(x)\right)$$

Dual Form of Wasserstein-2 distance

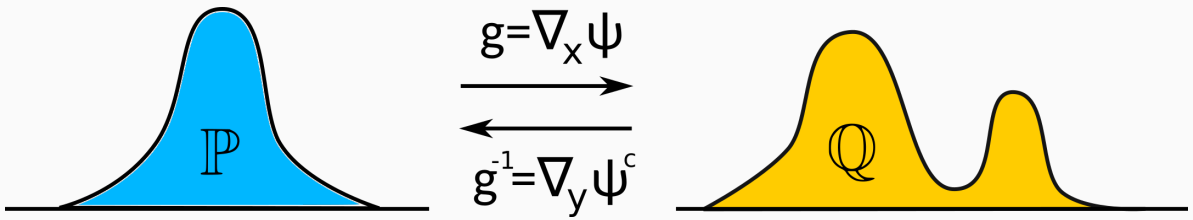
$$W_2(\mathbb{P},\mathbb{Q}) = -\min_{\psi\in \text{Conv}} \Big[\underbrace{\int_{\mathcal{X}} \psi(x)d\mathbb{P}(x) + \int_{\mathcal{Y}} \psi^c(y)d\mathbb{Q}(y)}_{\text{Corr}(\mathbb{P},\mathbb{Q})}\Big] + \text{Const}(\mathbb{P},\mathbb{Q})$$

$$\psi^c(y) = \max_x \big(\langle x, y\rangle - \psi(x)\big)$$

Important Fact

Optimal Convex Discriminator $\psi \Leftrightarrow$ Optimal Generator $g = \nabla_x \psi(x)$

Generator $g = \nabla_x \psi(x) \Leftrightarrow$ Inverse Generator $g^{-1} = \nabla_y \psi^c$



Push Forward Theorem

Theorem

Let $\mathbb{P}, \mathbb{Q}$ be two continuous probability distributions on $\mathcal{X} = \mathcal{Y} = \mathbb{R}^D$ with finite second moments and positive density everywhere. Let $\psi : \mathcal{X} \to \mathbb{R}$ be the convex minimizer of $\mathsf{Corr}(\mathbb{P}, \mathbb{Q})$*, i.e.*

$$\psi^* = \underset{\psi \in \mathsf{Conv}}{\arg\min} \left[\int_{\mathcal{X}} \psi(x) d\mathbb{P}(x) + \int_{\mathcal{Y}} \overline{\psi}(y) d\mathbb{Q}(y) \right].$$

*Let $\psi^\dagger : \mathcal{X} \to \mathbb{R}$ be an ϵ-***approximate** *convex minimizer of* $\mathsf{Corr}(\mathbb{P}, \mathbb{Q})$*:*

$$\int \psi^\dagger(x) d\mathbb{P}(x) + \int \overline{\psi^\dagger}(y) d\mathbb{Q}(y) = \left[\int \psi^*(x) d\mathbb{P}(x) + \int \overline{\psi^*}(y) d\mathbb{Q}(y) \right] + \epsilon.$$

Assume that $\psi^\dagger$ is twice differentiable, has M-Lipschitz bijective gradient $\nabla\psi^\dagger$ and strictly positive definite Hessian $\nabla^2\psi^\dagger \succ 0$. Then

$$\mathbb{W}_2^2\big(g^\dagger \circ \mathbb{P}, \mathbb{Q}\big) = \mathbb{W}_2^2\big(\nabla\psi^\dagger \circ \mathbb{P}, \mathbb{Q}\big) \leq M\epsilon,$$

i.e. mapping $g^\dagger = \nabla\psi^\dagger$ pushes $\mathbb{P}$ to be $M\epsilon$-close to $\mathbb{Q}$.

Decoding Theorem

Theorem
Let $\mathbb{S}$ be the real data distribution on $\mathcal{S} \subset \mathbb{R}^K$. Let $u : \mathcal{S} \to \mathcal{Y} = \mathbb{R}^D$ be the encoder and $v : \mathcal{Y} \to R^K$ be L-Lipschitz decoder. Assume that encoded distribution $\mathbb{Q} = u \circ \mathbb{S}$ is continuous with finite second moment and everywhere positive density.

Then, under the setting of the Push-Forward Theorem, we have

$$\mathbb{W}_2(\underbrace{v \circ g^\dagger \circ \mathbb{P}}_{\substack{\text{Generated data} \\ \text{distribuon}}}, \mathbb{S}) \leq L\sqrt{M\epsilon} + \Big(\frac{1}{2}\underbrace{\mathbb{E}_{\mathbb{S}}\|s - v \circ u(s)\|_2^2}_{\substack{\text{Autoencoder's} \\ \text{reconstrution loss}}}\Big)^{\frac{1}{2}},$$

where $g^\dagger : \mathcal{X} \to \mathcal{Y}$ is the fitted generative mapping in the latent space.

The distribution distance of the combined model = the quality of the latent fit + the reconstruction loss of the auto-encoder

Optimization of Wasserstein-2 Distance

$$\underset{\theta,\omega}{\text{minimize}} \left(\left[\int_{\mathcal{X}} \psi_\theta(x) d\mathbb{P}(x) + \int_{\mathcal{Y}} \psi_\omega^c(y) d\mathbb{Q}(y) \right] + \lambda R(\theta, \omega) \right)$$

Optimization of Wasserstein-2 Distance

$$\underset{\theta,\omega}{\text{minimize}} \left(\left[\int_{\mathcal{X}} \psi_\theta(x) d\mathbb{P}(x) + \int_{\mathcal{Y}} \psi_\omega^c(y) d\mathbb{Q}(y) \right] + \lambda R(\theta, \omega) \right)$$

Discriminators

$\psi_\theta, \psi_\omega^c$ are Input Convex Neural Networks

Optimization of Wasserstein-2 Distance

$$\underset{\theta,\omega}{\text{minimize}} \left(\Big[\int_{\mathcal{X}} \psi_\theta(x) d\mathbb{P}(x) + \int_{\mathcal{Y}} \psi_\omega^c(y) d\mathbb{Q}(y) \Big] + \lambda R(\theta, \omega) \right)$$

Discriminators

$\psi_\theta, \psi_\omega^c$ are Input Convex Neural Networks

Regularization term

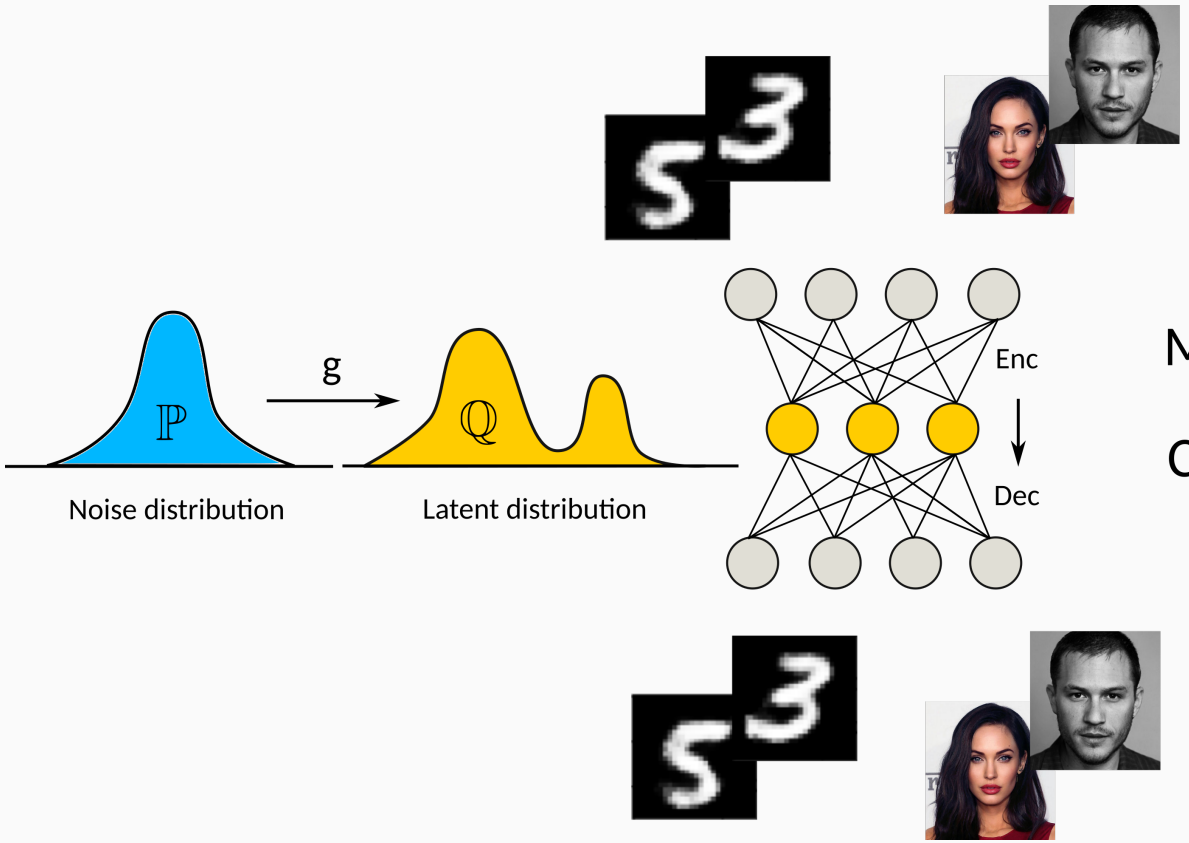
$$R(\theta, \omega) = (R_{\mathcal{X}}(\theta, \omega) + R_{\mathcal{Y}}(\theta, \omega))/2,$$

where for generative mappings $g_\theta = \nabla \psi_\theta$ and $g_\omega^{-1} = \nabla \psi_\omega^c$

$$R_{\mathcal{X}}(\theta, \omega) = \int_{\mathcal{X}} \|g_\omega^{-1} \circ g_\theta(x) - x\|^2 d\mathbb{P}(x),$$

$$R_{\mathcal{Y}}(\theta, \omega) = \int_{\mathcal{Y}} \|g_\theta \circ g_\omega^{-1}(y) - y\|^2 d\mathbb{Q}(y),$$

Latent Space Optimal Transport

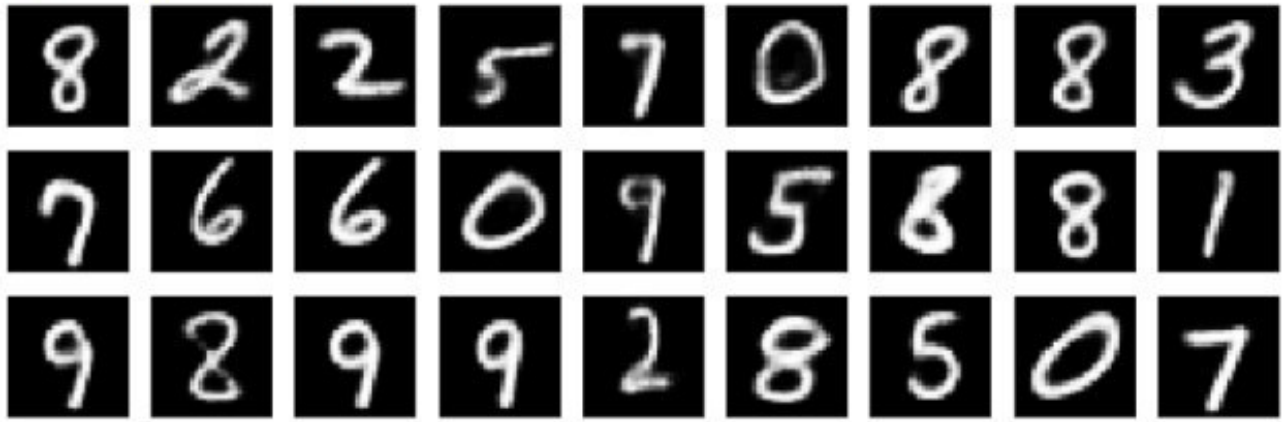


MNIST (AE, 16)

CelebA (ConvAE, 128)

MNIST Dataset

(a) W2-ICNN (Ours)

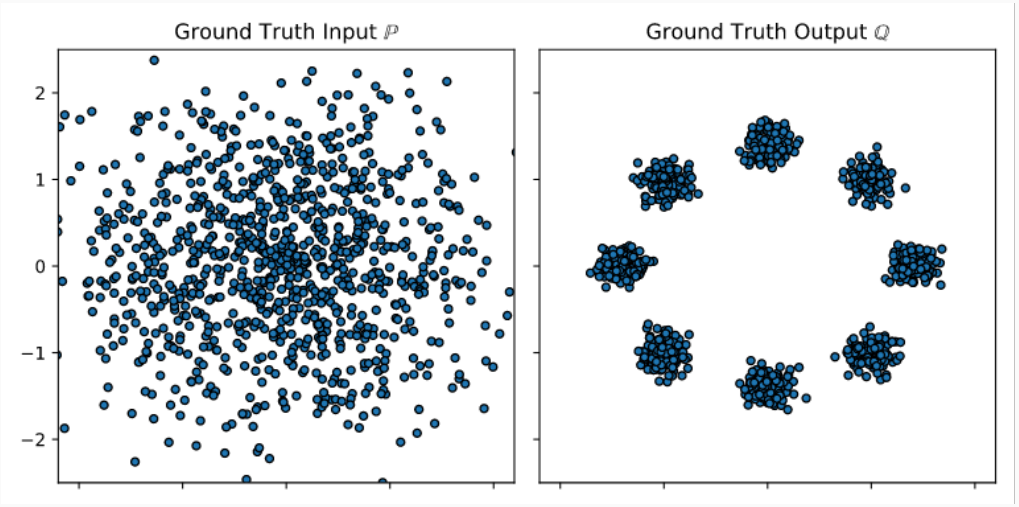


(b) Single Gaussian

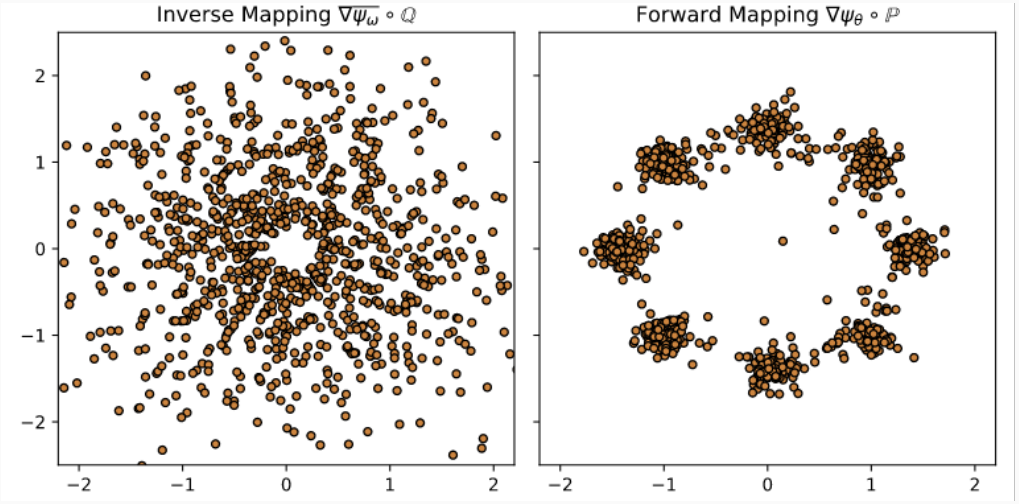
CelebA Dataset

W2-ICNN (Ours)

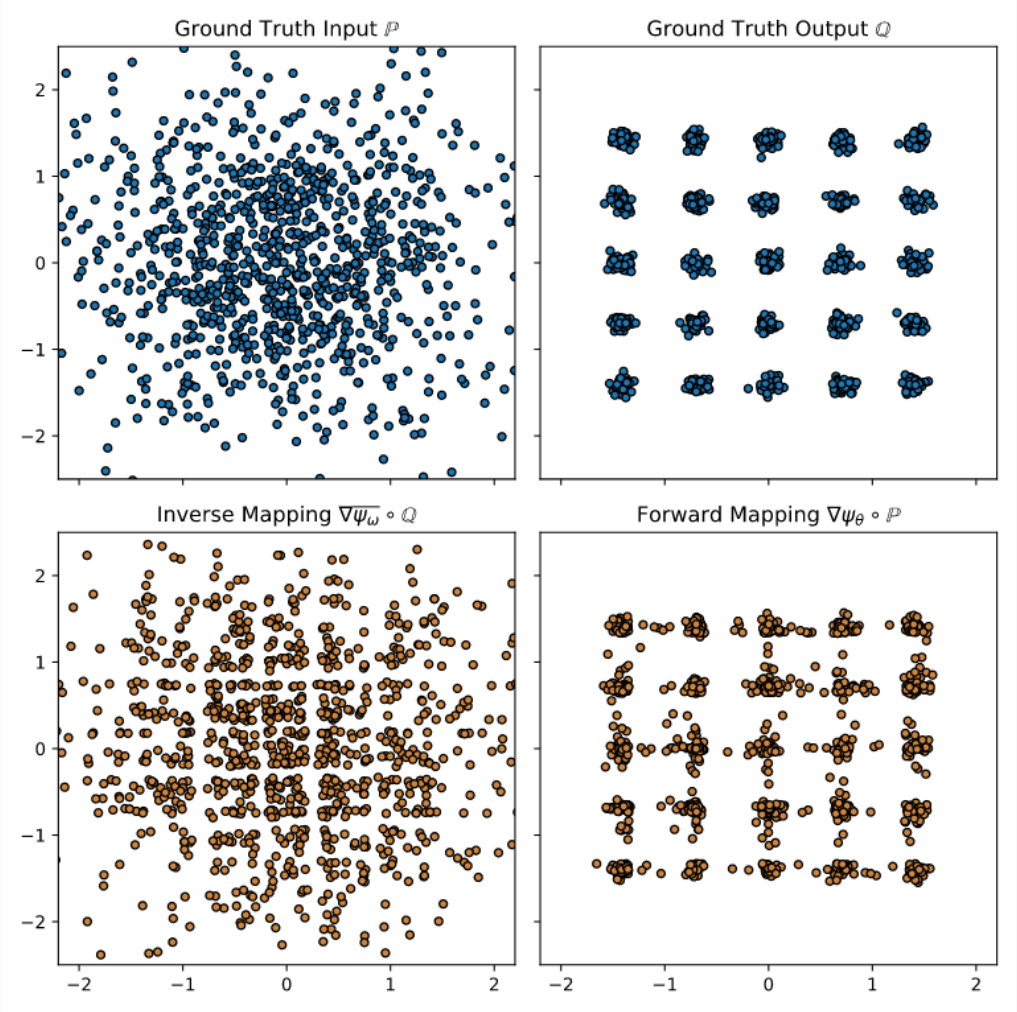
Distributions $\mathbb{P}$ (Gaussian) and $\mathbb{Q}$ (Mixture of 8 Gaussians)

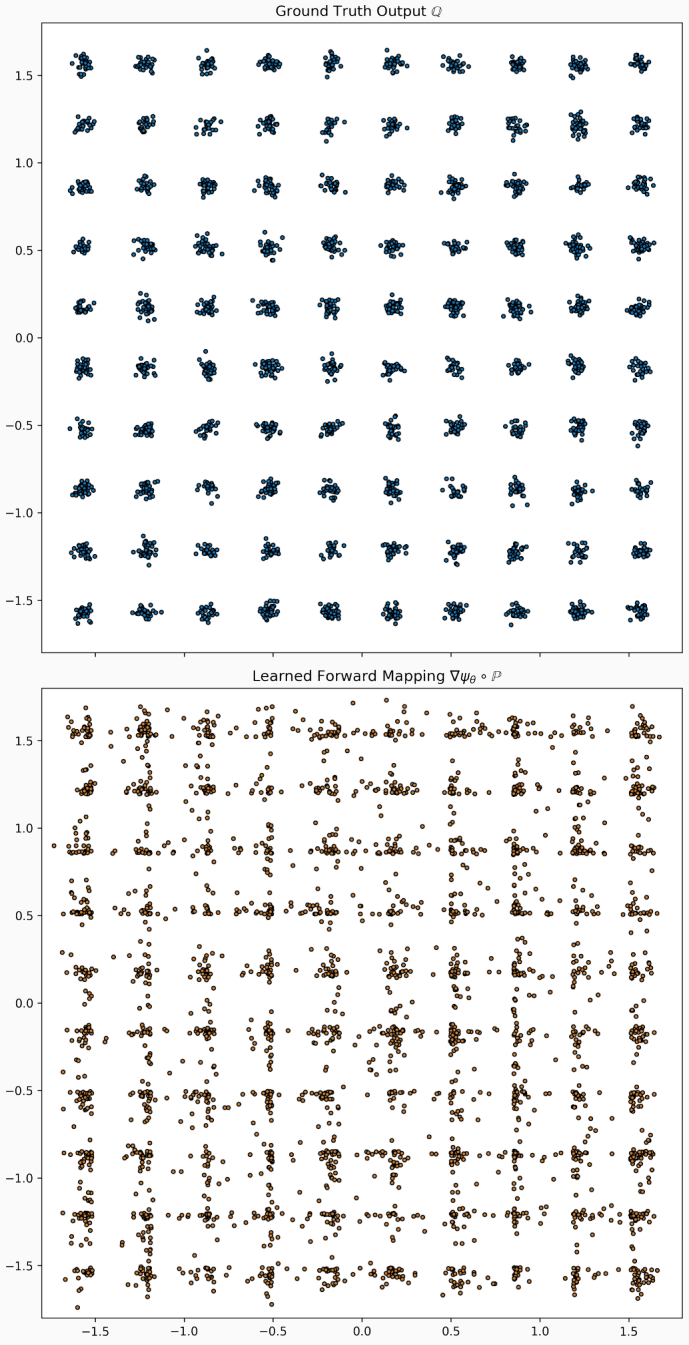


Distributions $\nabla\overline{\psi_\omega} \circ \mathbb{Q}$ and $\nabla\psi_\theta \circ \mathbb{P}$ after 30000 iterations.



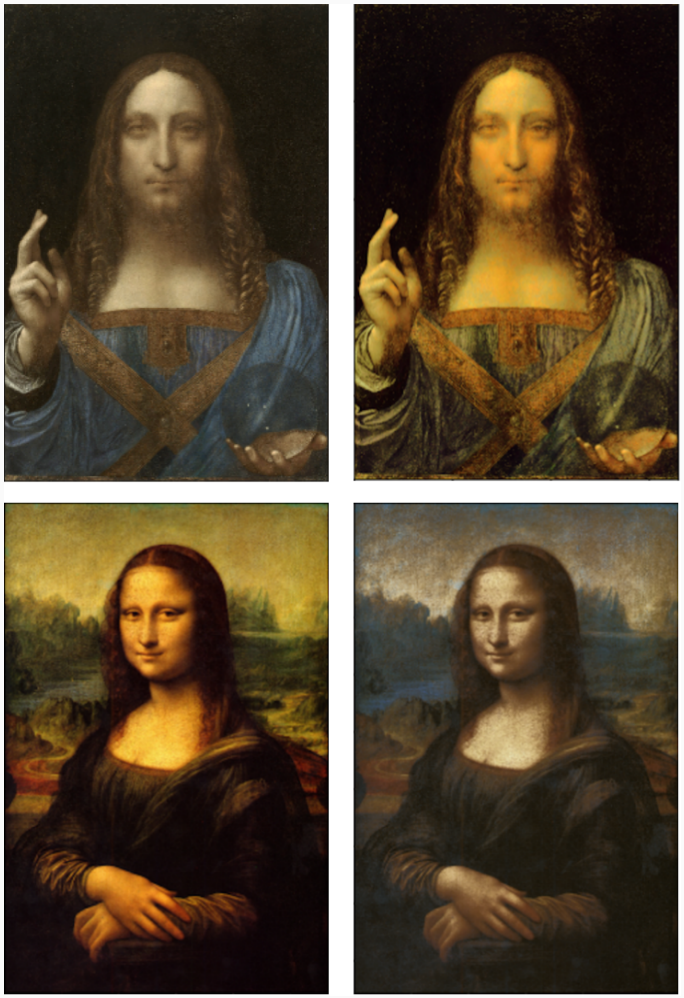
Gaussian distribution $\mathbb{P}$ (top left), Mixture $\mathbb{Q}$ of 25 Gaussians (top right) and distributions $\nabla\overline{\psi_\omega} \circ \mathbb{Q} \approx \mathbb{P}$ (bottom left), $\nabla\psi_\theta \circ \mathbb{P} \approx \mathbb{Q}$ (bottom right) fitted by our algorithm



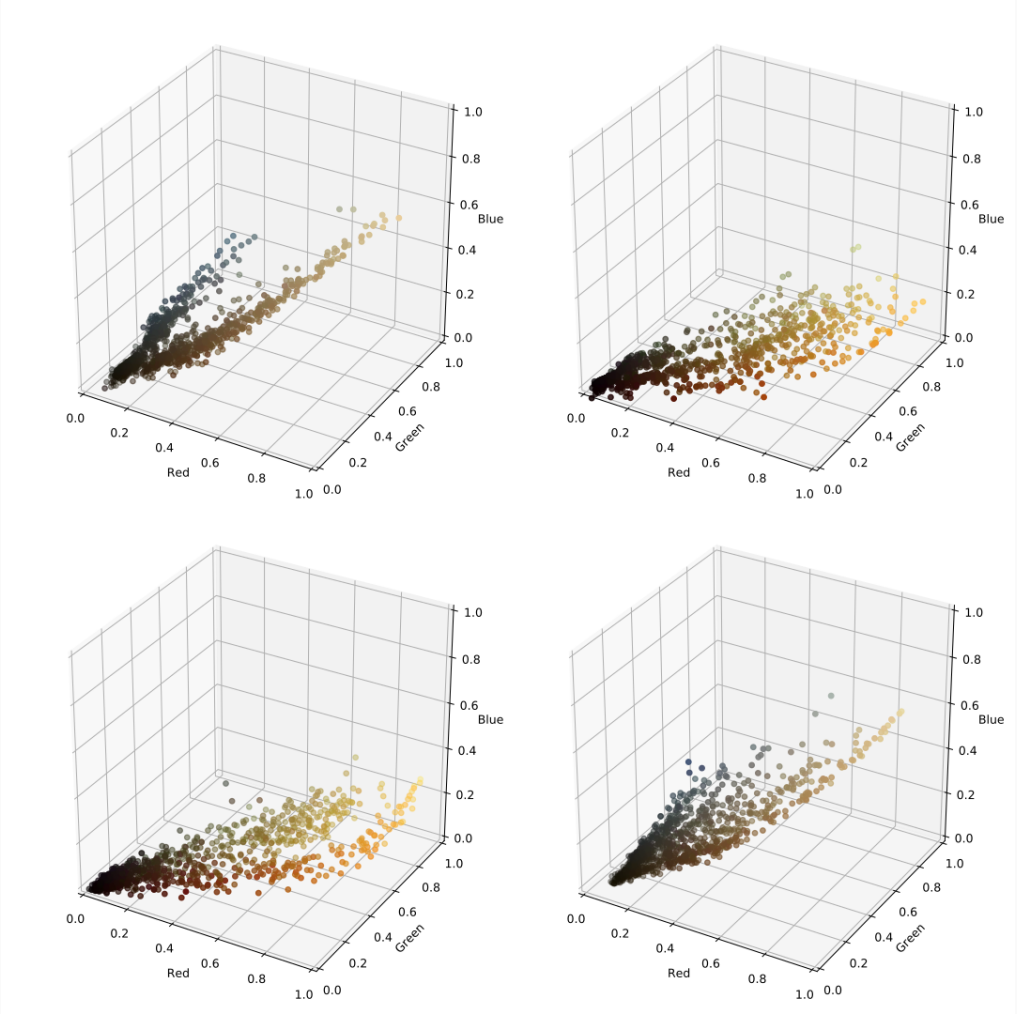


Mixture of 100 Gaussians $\mathbb{Q}$ and distribution $\nabla\psi_\theta \circ \mathbb{P}$ fitted by our

Color Transfer



Original images (on the left) and images obtained by color transfer (on the right). The sizes of images are 3300×4856 (on the top) and 2835×4289 (on the bottom)



Color palettes (3000 random pixels, best viewed in color) for the original images (on the left) and for images with transferred color (on the right)

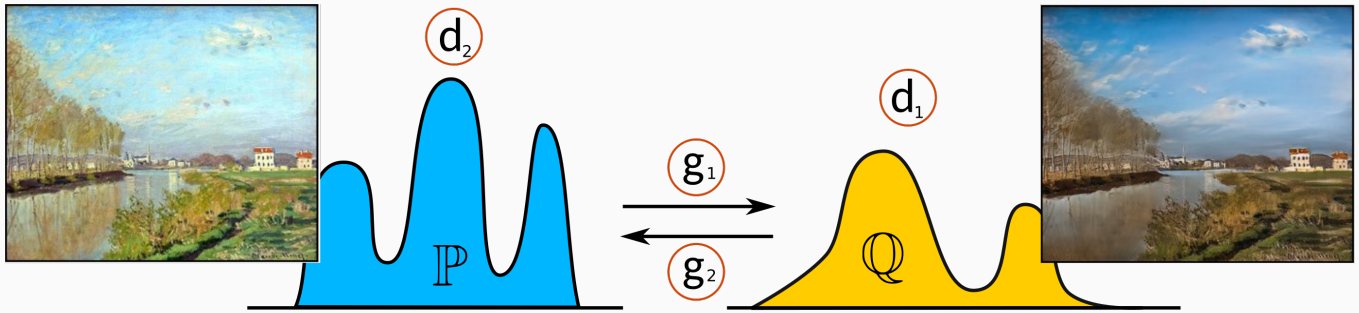
Domain Adaptation

	USPS→MNIST	MNIST→USPS
Target features	76.1%	85%
Mapped features	83.7%	85%

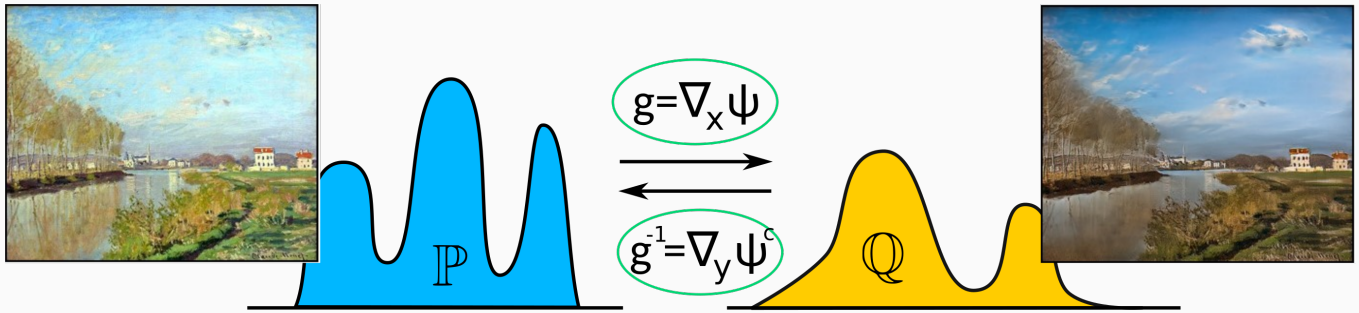
Table 1: 1-NN classification accuracy on MNIST $\rightleftarrows$ USPS domain adaptation problem.

Double Network Cycle GAN

Original Cycle GAN[6] - 4 networks



Wasserstein-2 Cycle GAN - 2 networks



[6] Jun-Yan Zhu et al. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232.

W2 Cycle GAN - Results

W2 Cycle GAN - Results

W2 Cycle GAN - Results

Wasserstein-2 Generative Nets - Conclusion

Key Advantages (w.r.t. classical GANs)

1. Non-minimax optimization!
   - Two "friendly" discriminators;
   - No discriminator-generator "fight": discriminator produces generator;
2. Superior properties
   - Monotonicity
   - Uniqueness
   - Existence
   - Invertibility
3. Extreme ICNN sparsity: up to 90% network weights vanish.