The Future of Deep Learning: Deep Generative Modeling

Jakub Tomczak

Amsterdam, April 5, 2019

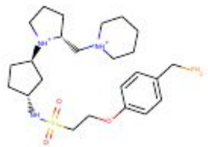
Applications of Deep Learning

“ i want to talk to you . ”
“i want to be with you . ”
“i do n’t want to be with you . ”
i do n’t want to be with you .
she did n’t want to be with him .

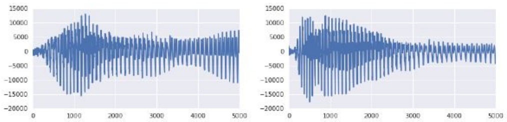
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

Text analysis

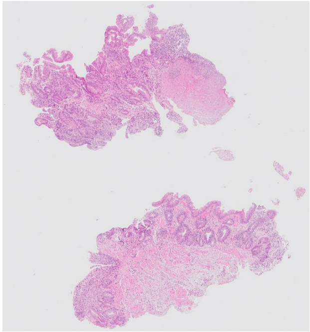
Image analysis



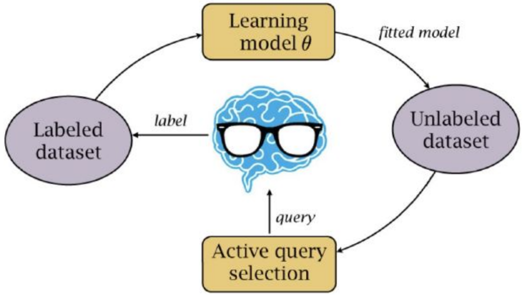
Graph analysis



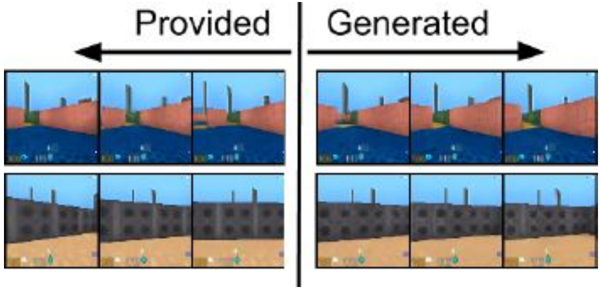
Audio analysis



Medical data



Active Learning



Reinforcement Learning

and more...

Is generative modeling important?

We learn a neural network to classify images:

Is generative modeling important?

We learn a neural network to classify images:

p(**panda**|x)=0.99

...

Is generative modeling important?

We learn a neural network to classify images:

$+$

p(**panda**|x)=0.99

...

noise

Is generative modeling important?

We learn a neural network to classify images:

+

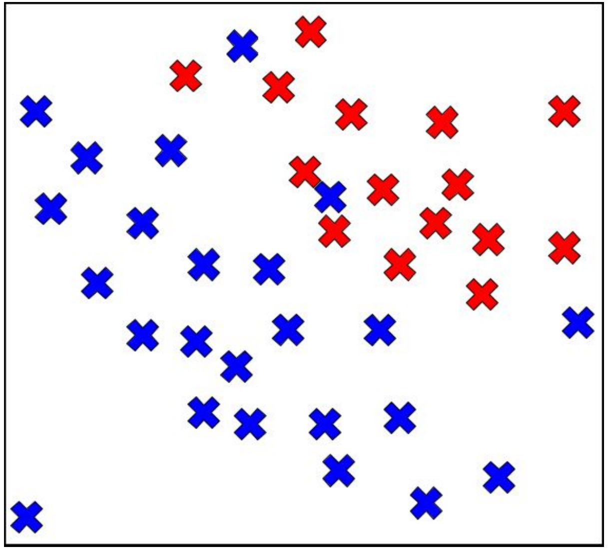
=

p(**panda**|x)=0.99
...

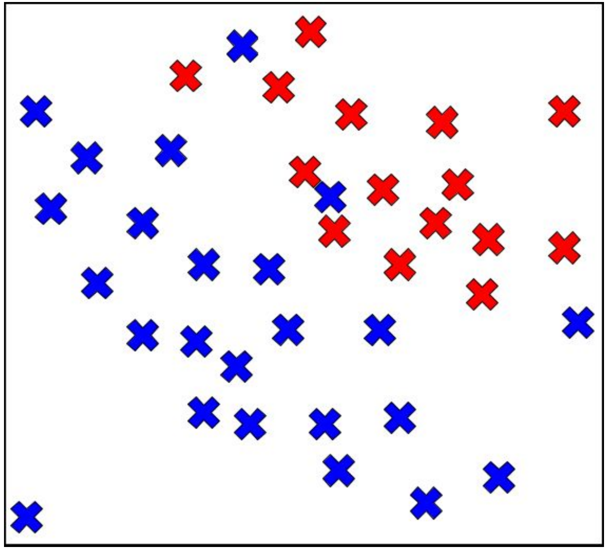
noise

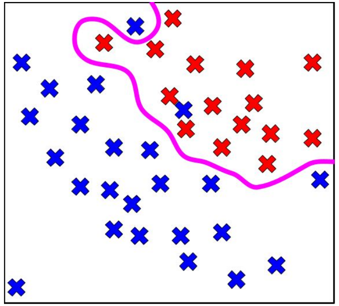
p(**panda**|x)=0.01
…
p(**monkey**|x)=0.9

Is generative modeling important?



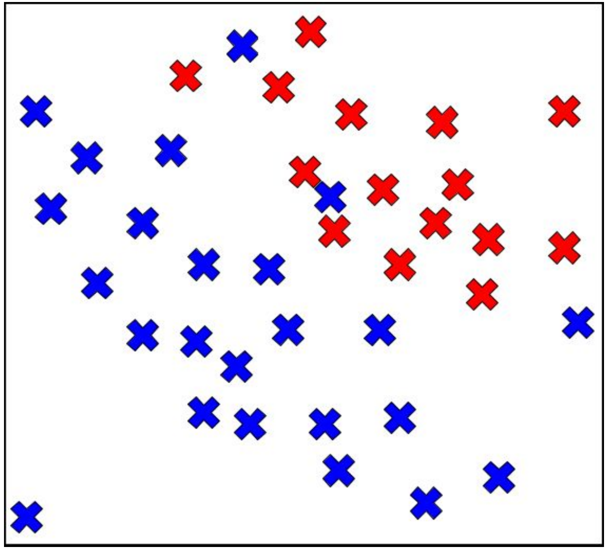
Is generative modeling important?

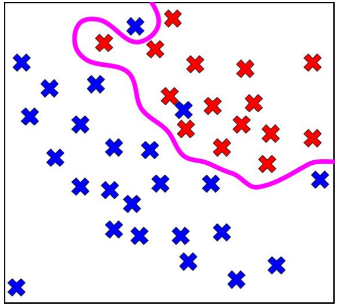




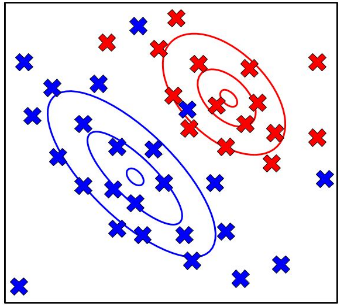
$p_\theta(y|x)$

Is generative modeling important?



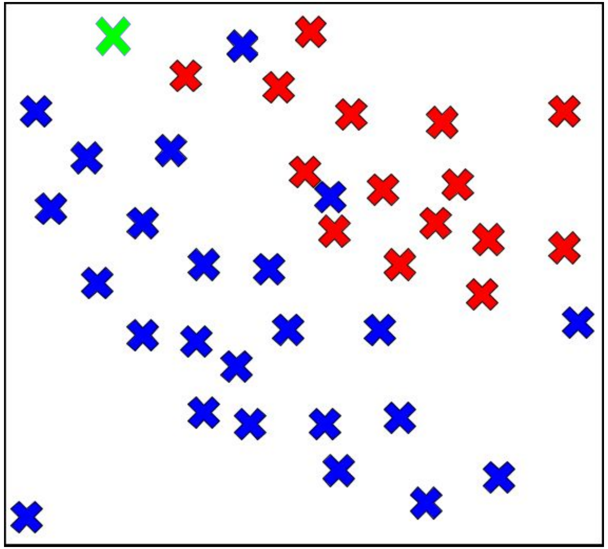


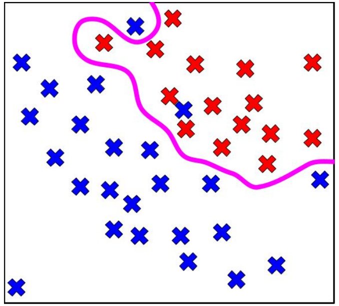
$p_\theta(y|x)$



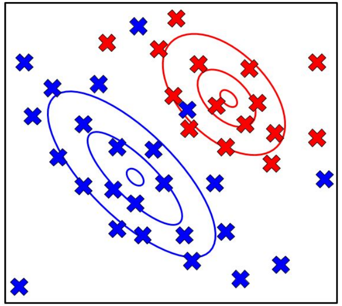
$p_\theta(x, y) = p_\theta(y|x)\ p_\theta(x)$

Is generative modeling important?





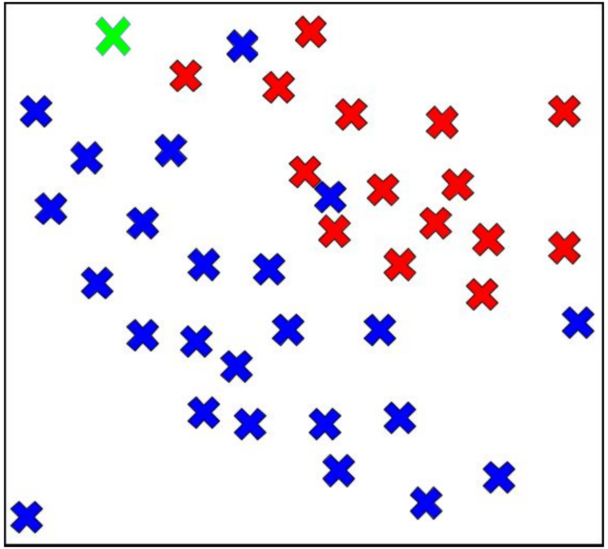
$$p_\theta(y|x)$$

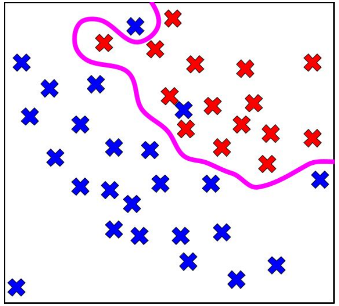


$$p_\theta(x,y) = p_\theta(y|x)\ p_\theta(x)$$

Is generative modeling important?

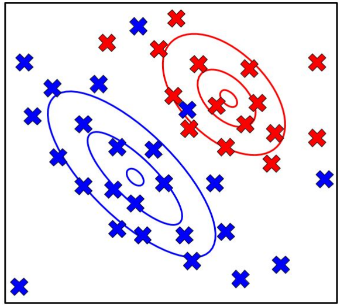
new data





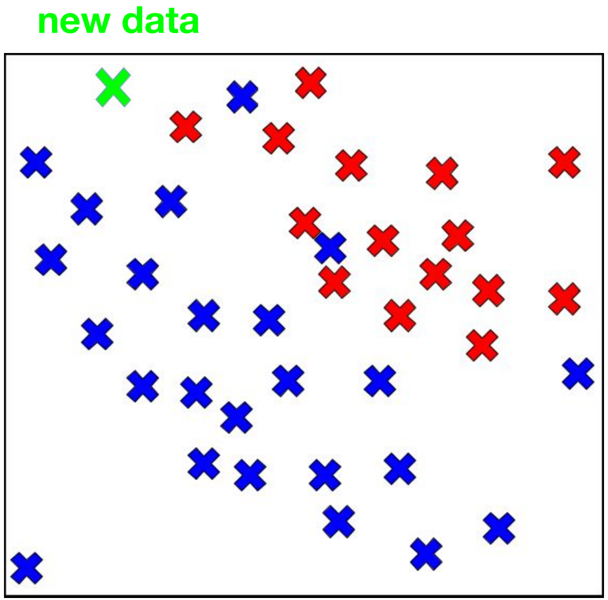
$$p_\theta(y|x)$$

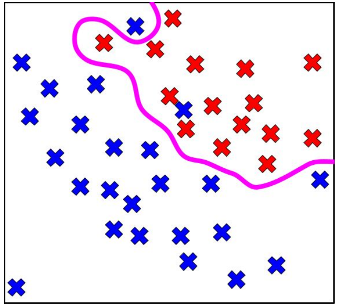
High probability
of the **blue** label.
=
Highly probable decision!



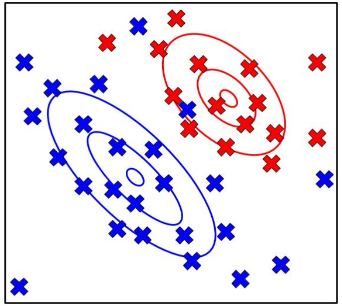
$$p_\theta(x, y) = p_\theta(y|x)\ p_\theta(x)$$

Is generative modeling important?





$$p_\theta(y|x)$$

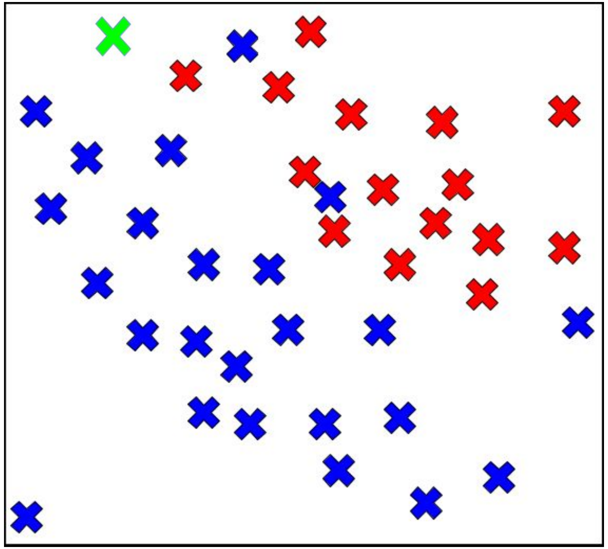


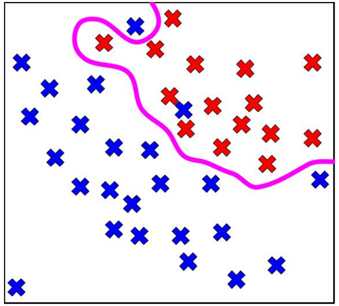
$$p_\theta(x, y) = p_\theta(y|x)\ p_\theta(x)$$

High probability
of the **blue** label.
=
Highly probable decision!

High probability
of the **blue** label.
x
Low probability
of the **object**
=
Uncertain decision!

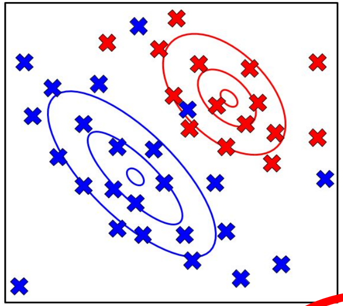
Is generative modeling important?





$$p_\theta(y|x)$$

High probability
of the **blue** label.
=
Highly probable decision!



$$p_\theta(x, y) = p_\theta(y|x)\, p_\theta(x)$$

High probability
of the **blue** label.
x
Low probability
of the **object**
=
Uncertain decision!

!

What is generative modeling about?

Understanding:

$$p(y, \mathbf{x}) = p(y|\mathbf{x})\ p(\mathbf{x})$$

finding underlying factors (**discovery**)

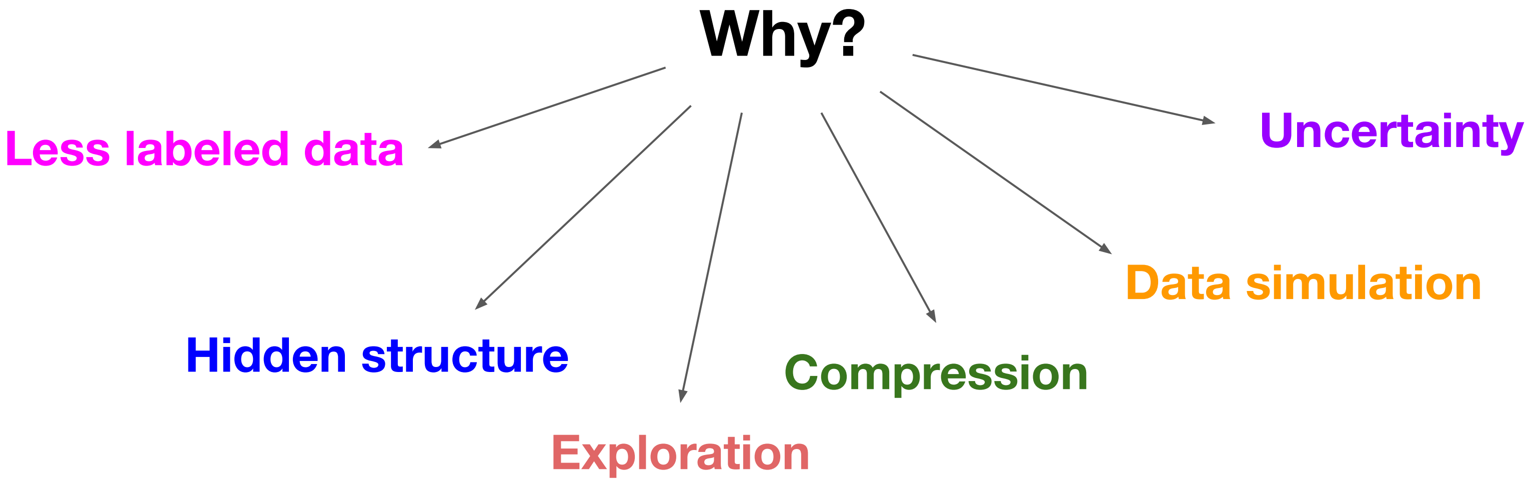
predicting and anticipating future events (**planning**)

finding analogies (**transfer learning**)

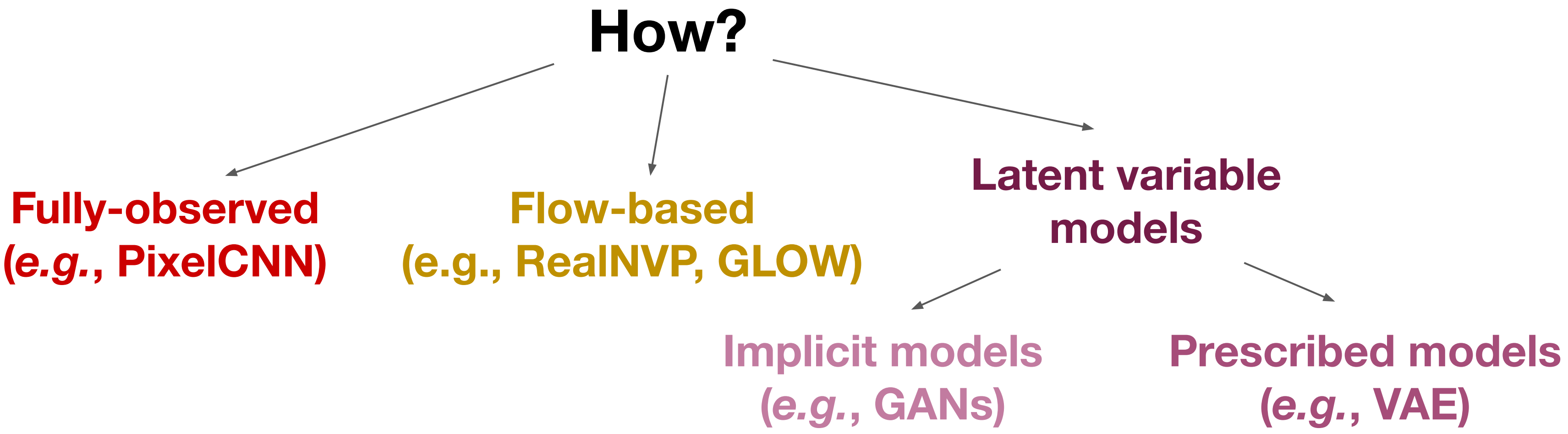
detecting rare events (**anomaly detection**)

decision making

Why generative modeling?



Generative modeling: **How**?

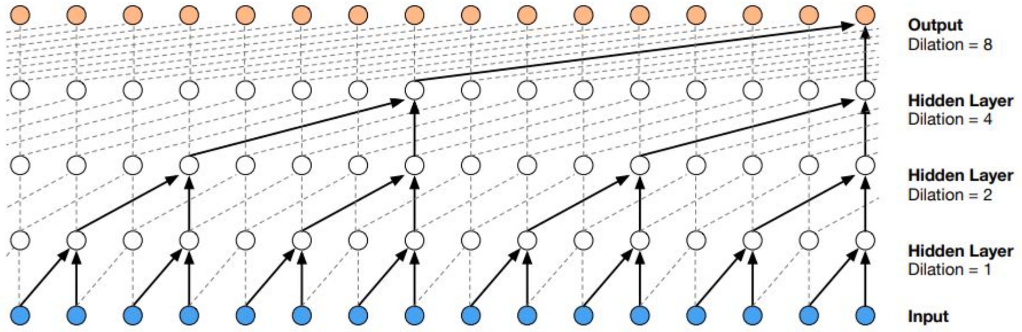


Generative modeling: **Auto-regressive models**

General idea is to factorise the joint distribution:

$$p(\mathbf{x}) = p(x_1) \prod_{d=2}^{D} p(x_d | \mathbf{x}_{1:d-1})$$

and use neural networks (e.g., convolutional NN) to model it efficiently:



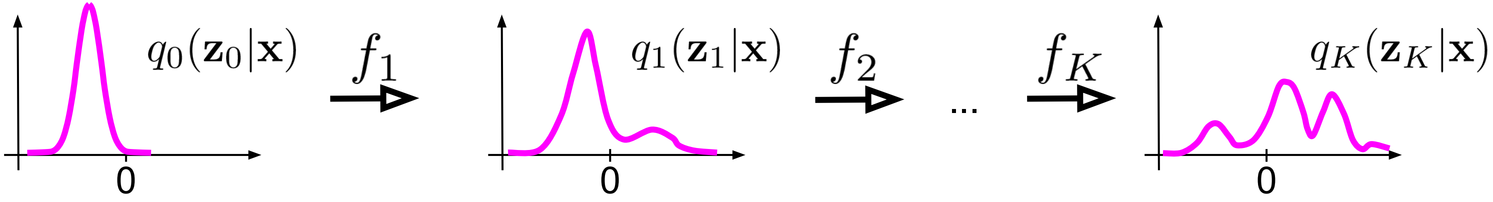
Van Den Oord, A., et al. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Generative modeling: **Flow-based models**

- Sample from a “simple” distribution:

$$\mathbf{z}_0 \sim q_0(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|\mu(\mathbf{x}), \mathrm{diag}(\sigma^2(\mathbf{x}))\right)$$

- Apply a sequence of *K* **invertible** transformations: $f_k : \mathbb{R}^M \to \mathbb{R}^M$



and the change of variables yields:

$$q_K(\mathbf{z}_K|\mathbf{x}) = q_0(\mathbf{z}_0|\mathbf{x}) \prod_{k=1}^{K} \left|\det \frac{\partial f_k(\mathbf{z}_{k-1})}{\partial \mathbf{z}_{k-1}}\right|^{-1}$$

Rezende, D. J., & Mohamed, S. (2015). Variational inference with normalizing flows. ICML 2015

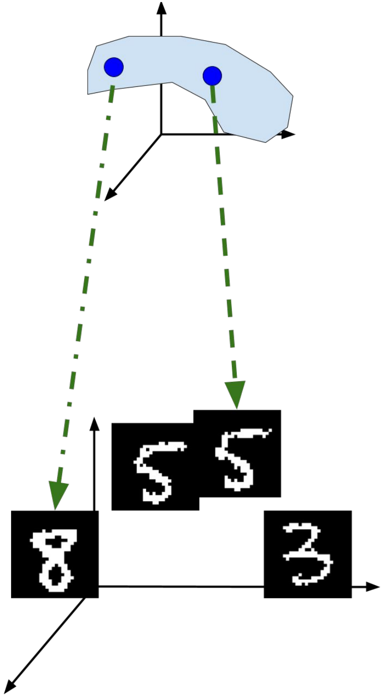
Generative modeling: **Latent Variable Models**

We assume data lies on a low-dimensional manifold so the generator is:

$$\mathbf{x} = f_\theta(\mathbf{z})$$

where:

$$\mathbf{x} \in \mathcal{X} \ (\text{e.g. } \mathcal{X} = \mathbb{R}^D\ ) \quad \text{and } \mathbf{z} \in \mathbb{R}^d$$



Two main approaches:

→ **Generative Adversarial Networks** (GANs)

→ **Variational Auto-Encoders** (VAEs)

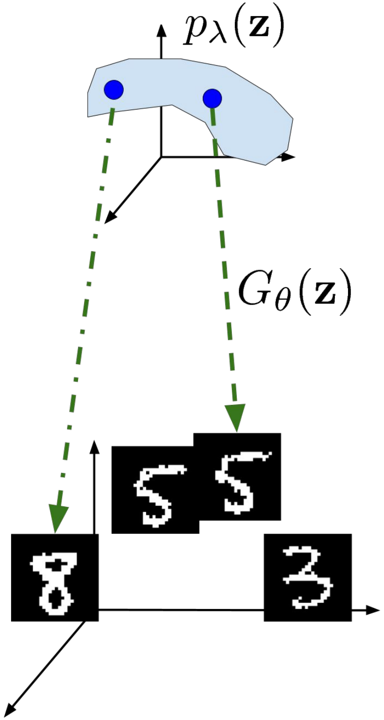
Generative modeling: **GANs**

We assume a **deterministic generator**:

$$\mathbf{x} = G_\theta(\mathbf{z})$$

and **a prior** over latent space:

$$\mathbf{z} \sim p_\lambda(\mathbf{z})$$



Generative modeling: **GANs**

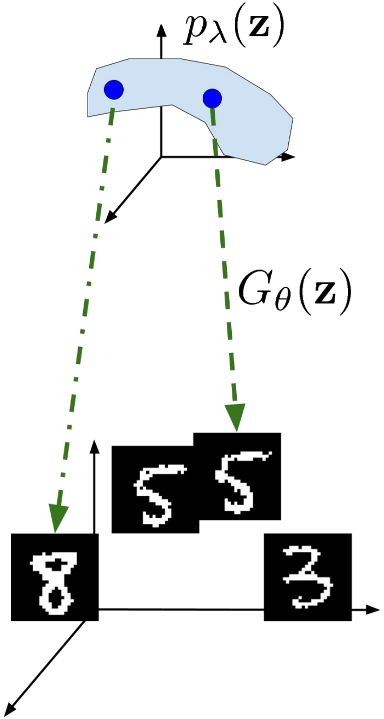
We assume a **deterministic generator**:

$$\mathbf{x} = G_\theta(\mathbf{z})$$

and **a prior** over latent space:

$$\mathbf{z} \sim p_\lambda(\mathbf{z})$$

How to train it?



Generative modeling: **GANs**

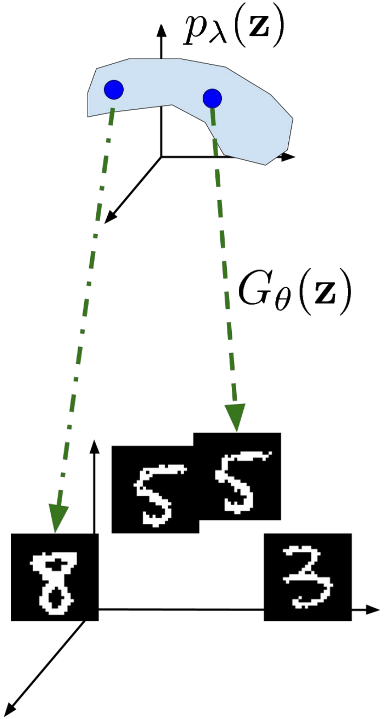
We assume a **deterministic generator**:

$$\mathbf{x} = G_\theta(\mathbf{z})$$

and **a prior** over latent space:

$$\mathbf{z} \sim p_\lambda(\mathbf{z})$$

How to train it? By using a game!



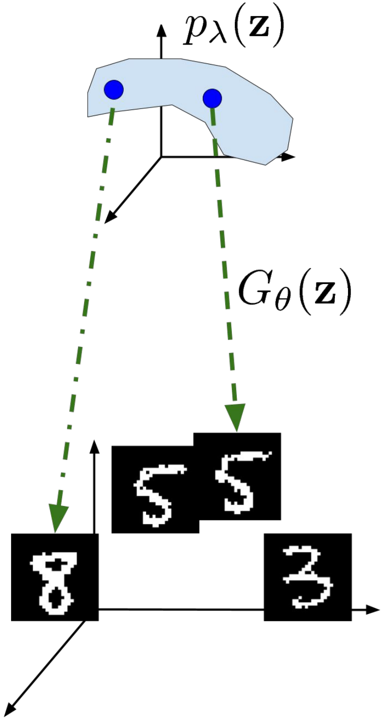
Generative modeling: **GANs**

We assume a **deterministic generator**:

$$\mathbf{x} = G_\theta(\mathbf{z})$$

and **a prior** over latent space:

$$\mathbf{z} \sim p_\lambda(\mathbf{z})$$



How to train it? By using a game!

For this purpose, we assume a discriminator:

$$D_\psi(\mathbf{x}) \in [0, 1]$$

Generative modeling: **GANs**

The learning process is as follows:

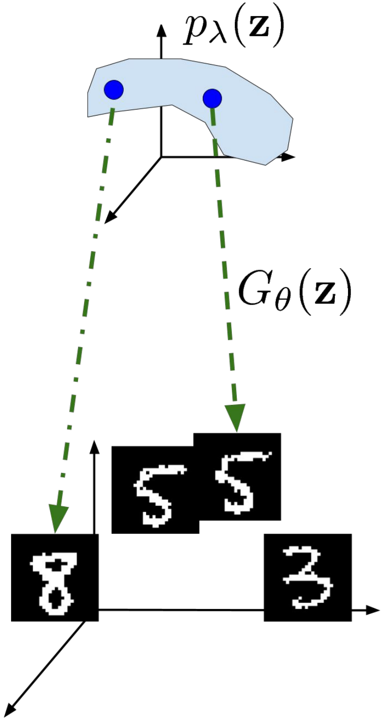
→ the generator tries to **fool** the discriminator;

→ the discriminator tries to **distinguish** between the real and fake images.

We define the learning problem as a min-max problem:

$$\min_{\theta} \max_{\psi} \mathbb{E}_{\mathbf{x} \sim p_{data}} \Big[\ln D_{\psi}(\mathbf{x}) \Big] - \mathbb{E}_{\mathbf{z} \sim p_{\lambda}(\mathbf{z})} \Big[\ln \big(1 - D_{\psi}(G(\mathbf{z})) \big) \Big]$$

In fact, we have a **learnable loss** function!



Goodfellow, I., et al. (2014). Generative adversarial nets. *NIPS 2014*

Generative modeling: **GANs**

The learning process is as follows:

→ the generator tries to **fool** the discriminator;

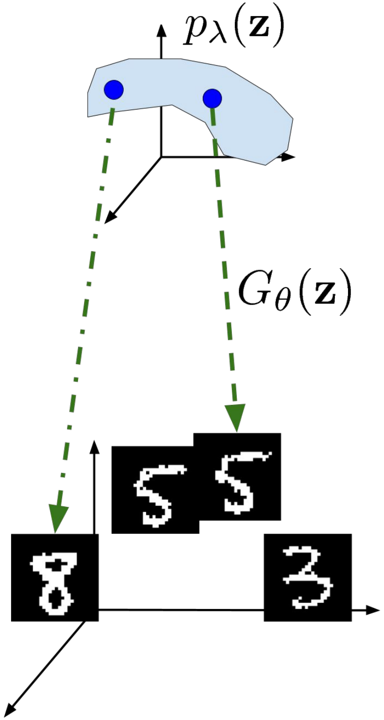
→ the discriminator tries to **distinguish** between the real and fake images.

We define the learning problem as a min-max problem:

$$\min_{\theta} \max_{\psi} \mathbb{E}_{\mathbf{x}\sim p_{data}} \Big[\ln D_{\psi}(\mathbf{x}) \Big] - \mathbb{E}_{\mathbf{z}\sim p_{\lambda}(\mathbf{z})} \Big[\ln \big(1 - D_{\psi}(G(\mathbf{z})) \big) \Big]$$

In fact, we have a **learnable loss** function!

→It learns high-order statistics.



Goodfellow, I., et al. (2014). Generative adversarial nets. *NIPS 2014*

Generative modeling: **GANs**

Pros:
→ we don't need to specify a likelihood function;
→ very flexible;
→ the loss function is trainable;
→ perfect for data simulation.

Cons:
→ we don't know the distribution;
→ training is highly unstable (min-max objective);
→ missing mode problem.

Generative modeling: **VAEs**

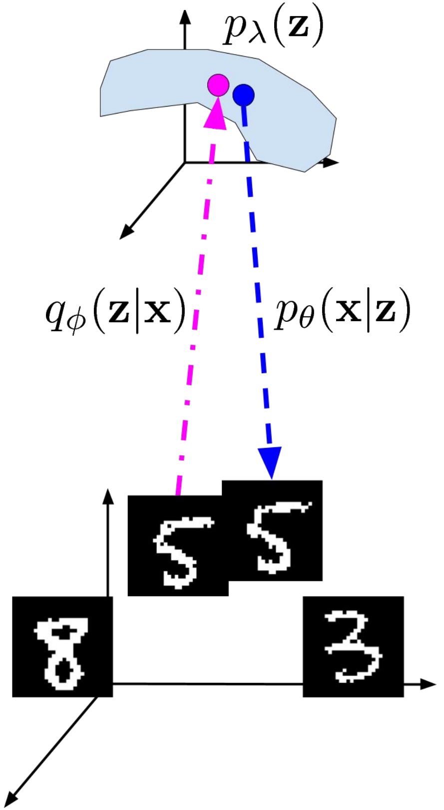
We assume **a stochastic generator** (decoder) and a **prior**:

$$\mathbf{z} \sim p_{\lambda}(\mathbf{z})$$

$$\mathbf{x} \sim p_{\theta}(\mathbf{x}|\mathbf{z})$$

Additionally, we use **a variational posterior** (encoder):

$$\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$$



How to train it? Using the log-likelihood function!

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_{\phi}(\mathbf{z}|\mathbf{x})}\Big[\ln p_{\theta}(\mathbf{x}|\mathbf{z})\Big] - \mathrm{KL}\Big[q_{\phi}(\mathbf{z}|\mathbf{x})||p_{\lambda}(\mathbf{z})\Big]$$

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. (*ICLR 2014*)

Variational Auto-Encoder: Extensions

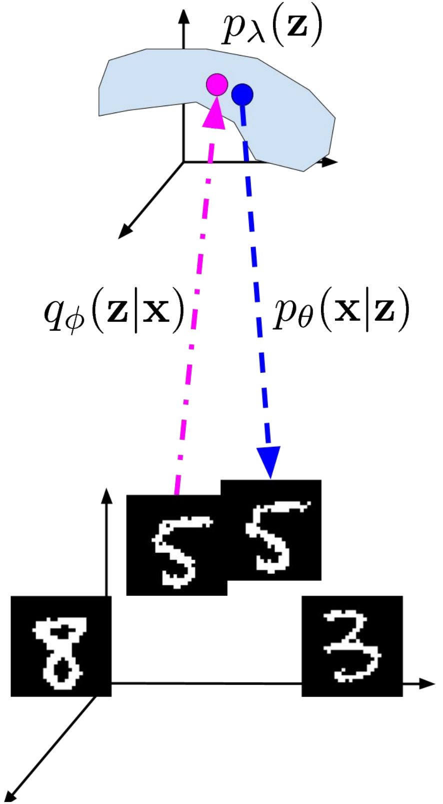
$$q_\phi(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}|\mathbf{z})\, p_\lambda(\mathbf{z})$$

Normalizing flows
Volume-preserving flows
non-Gaussian distributions

Fully-connected
ConvNets
PixelCNN
Other

Autoregressive Prior
Objective Prior
Stick-Breaking Prior
VampPrior

Importance Weighted AE
Renyi Divergence
Stein Divergence



Tomczak, J. M., & Welling, M. (2016). Improving variational auto-encoders using householder flow. *NIPS Workshop 2016.*
Berg, R. V. D., Hasenclever, L., **Tomczak, J. M.**, & Welling, M. (2018). Sylvester Normalizing Flows for Variational Inference. *UAI 2018*.
Tomczak, J. M., & Welling, M. (2017). VAE with a VampPrior. *arXiv preprint arXiv:1705.07120*. (*AISTATS 2018*)
Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., & **Tomczak, J. M.** (2018). Hyperspherical Variational Auto-Encoders. *UAI 2018.*

Generative modeling: **VAEs**

Pros:

→ we know the distribution and can calculate the likelihood function;

→ we can encode an object in a low-dim manifold (compression);

→ training is stable;

→ no missing modes.

Cons:

→ we need know the distribution;

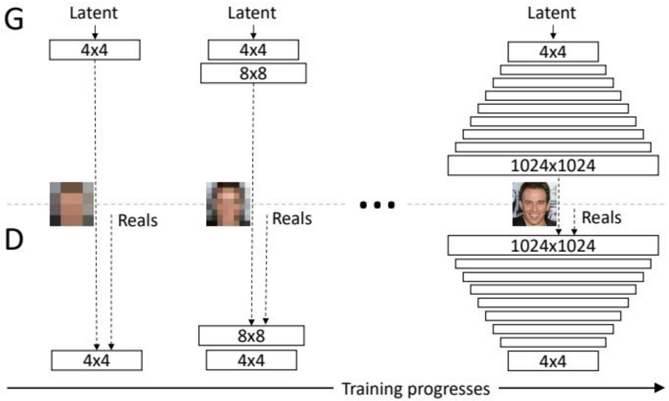
→ we need a flexible encoder and prior;

→ blurry images (so far…).

Recent successes: **Image generation (GANs)**

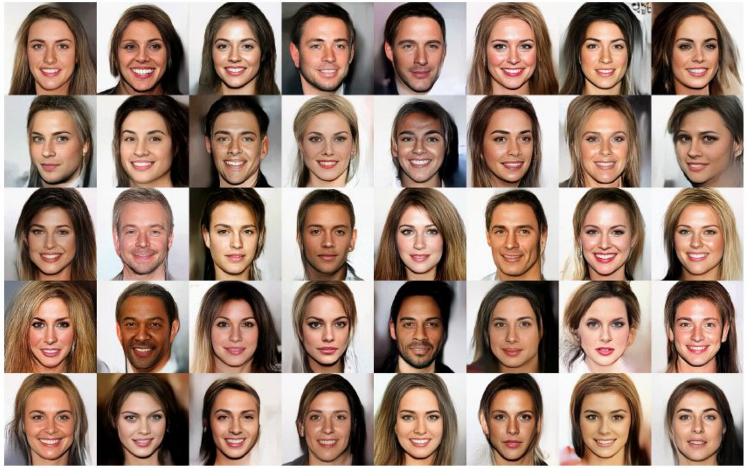
generated

real



Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *ICLR 2017*.

Recent successes: **Image generation (GLOW)**



generations

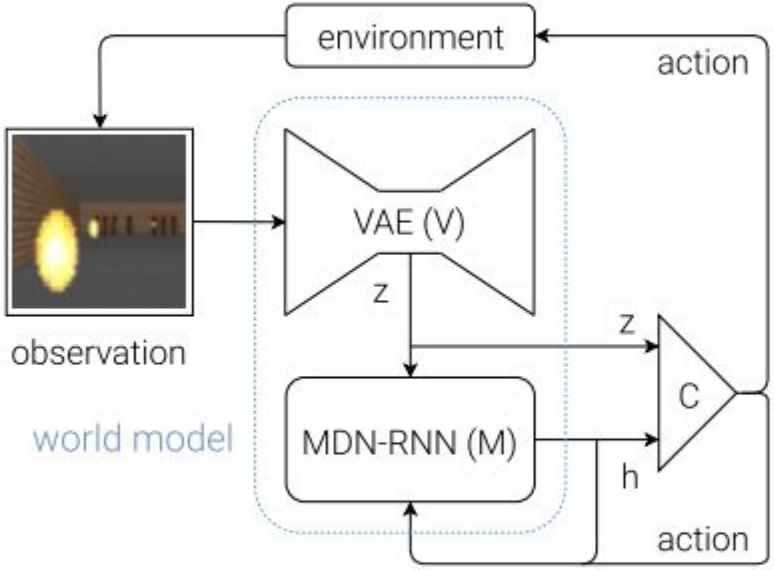
interpolations

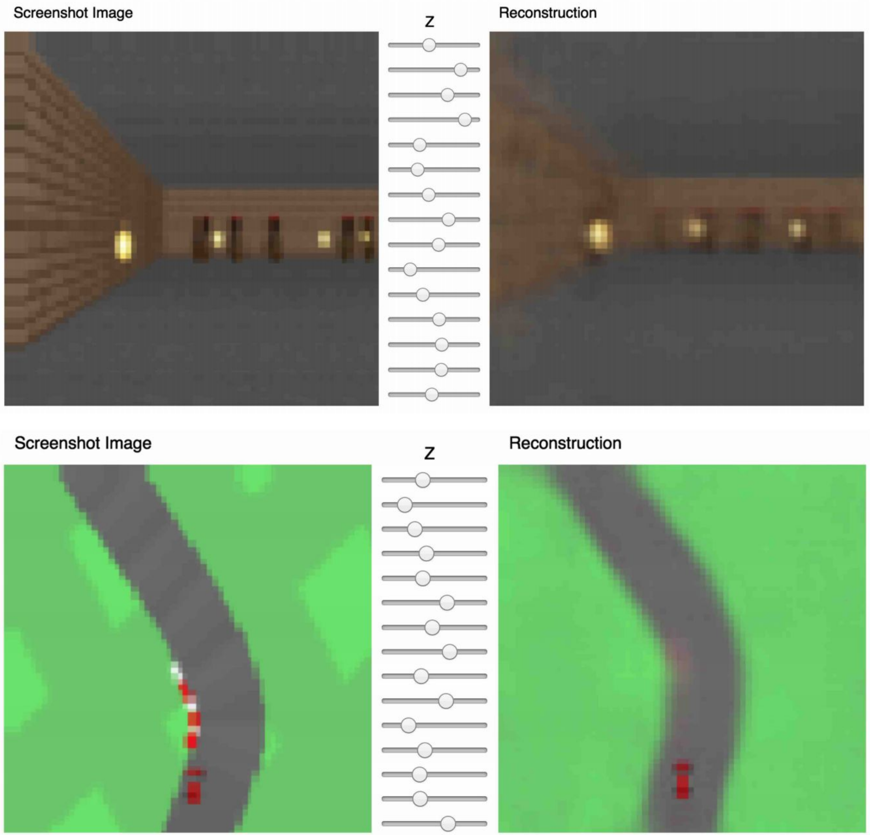
(a) Smiling (b) Pale Skin

(c) Blond Hair (d) Narrow Eyes

Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*

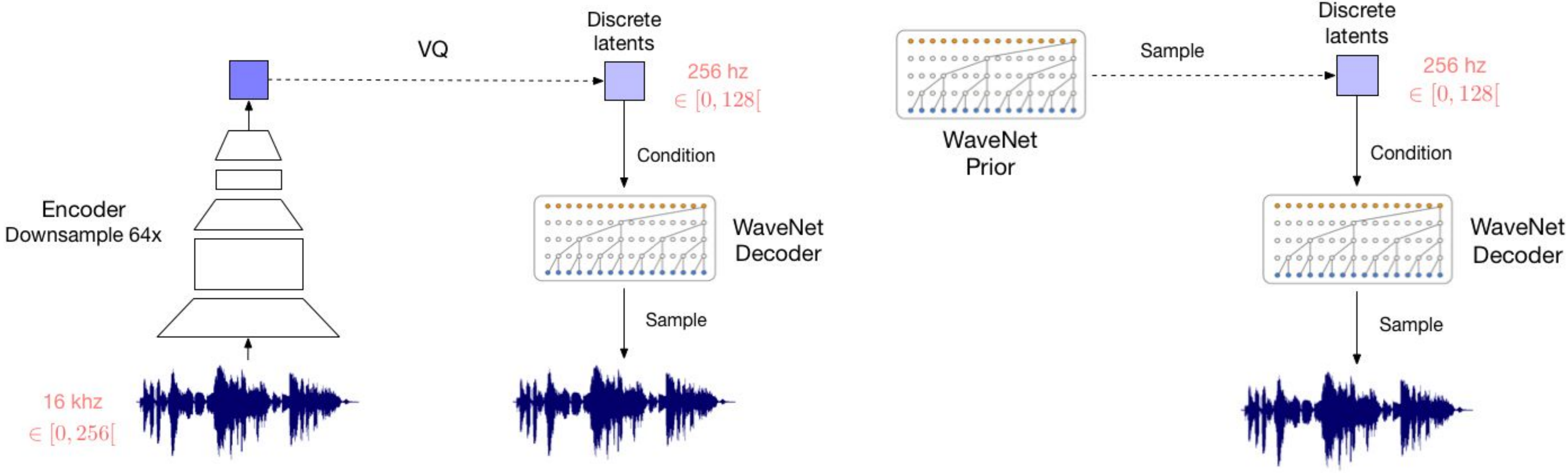
Recent successes: **Reinforcement learning**





Ha, D., & Schmidhuber, J. (2018). World models. arXiv preprint. *arXiv preprint arXiv:1803.10122*.

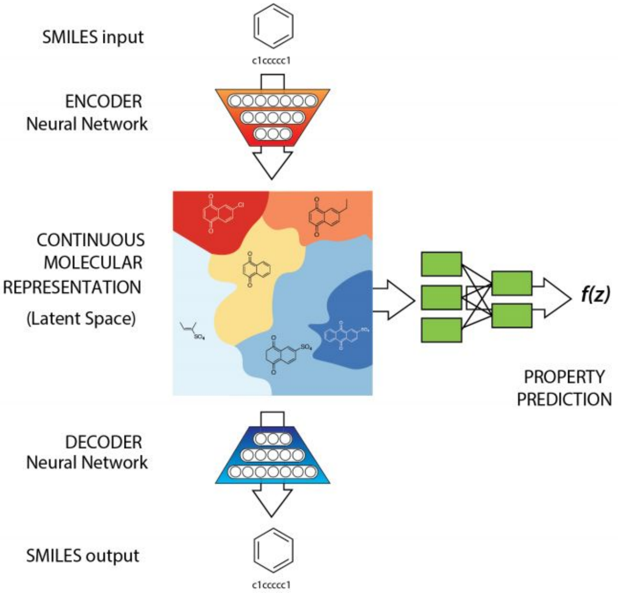
Recent successes: **Audio generation**

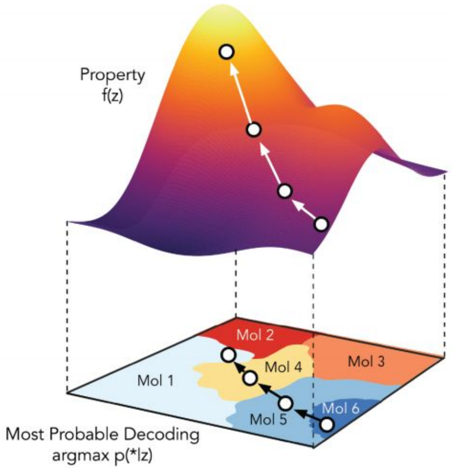


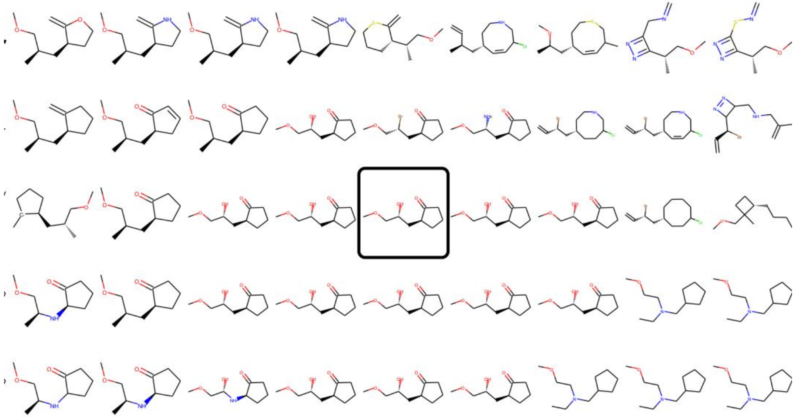
reconstruction **generation**

van den Oord, A., & Vinyals, O. (2017). Neural discrete representation learning. *NIPS 2017*.

Recent successes: **Drug discovery**



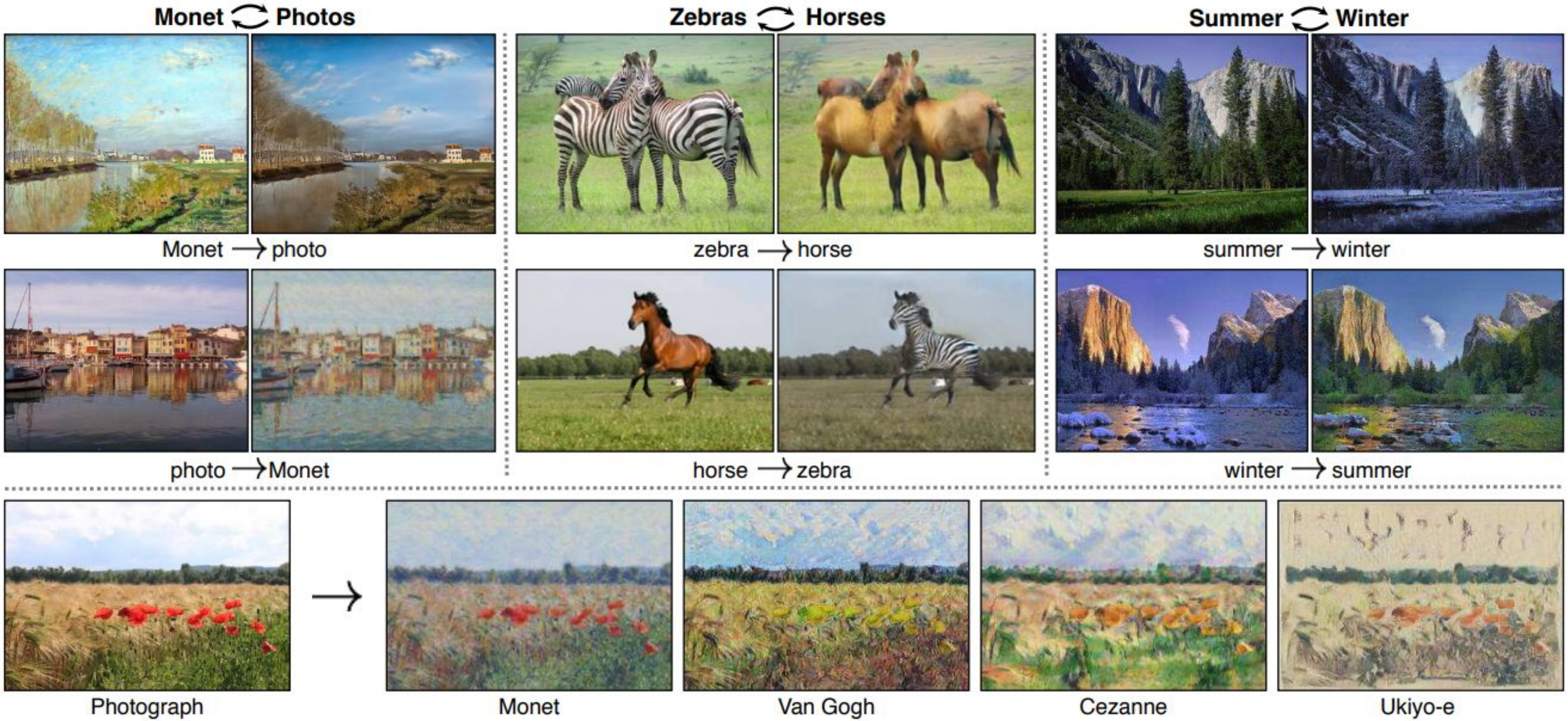




Gómez-Bombarelli, R., et al. (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules ACS Cent.

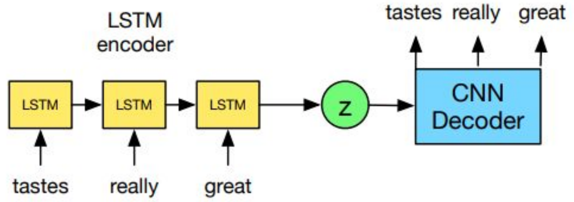
Kusner, M. J., Paige, B., & Hernández-Lobato, J. M. (2017). Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*.

Recent successes: **Style transfer**

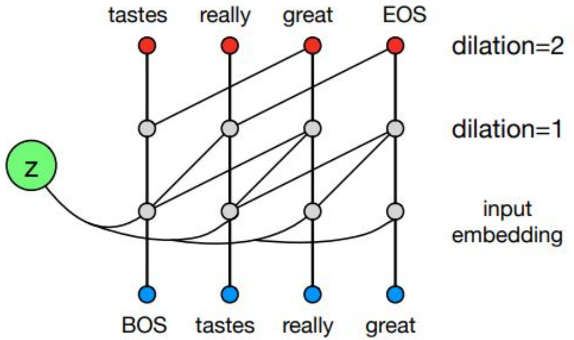


Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *CVPR 2017*.

Recent successes: **Text generation**

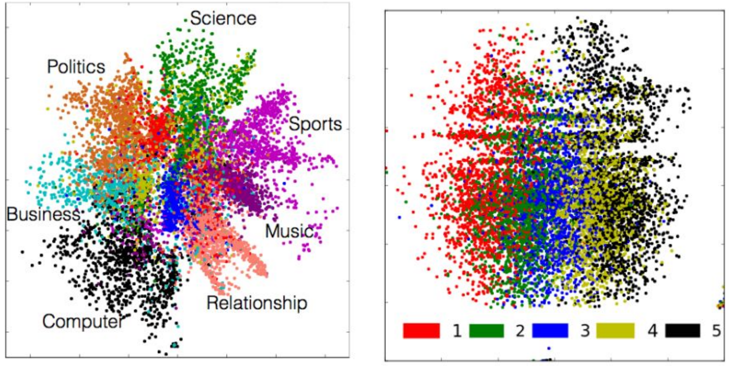


(a) VAE training graph using a dilated CNN decoder.



(b) Digram of dilated CNN decoder.

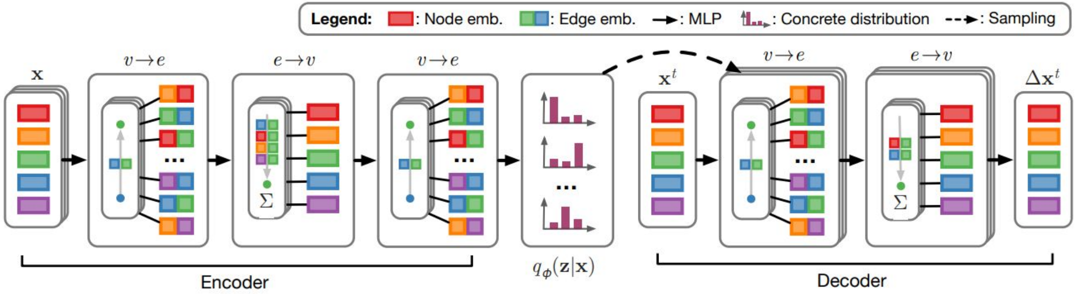
1 star	the food was good but the service was horrible . took forever to get our food . we had to ask twice for our check after we got our food . will not return .
2 star	the food was good , but the service was terrible . took forever to get someone to take our drink order . had to ask 3 times to get the check . food was ok , nothing to write about .
3 star	came here for the first time last night . food was good . service was a little slow . food was just ok .
4 star	food was good , service was a little slow , but the food was pretty good . i had the grilled chicken sandwich and it was really good . will definitely be back !
5 star	food was very good , service was fast and friendly . food was very good as well . will be back !

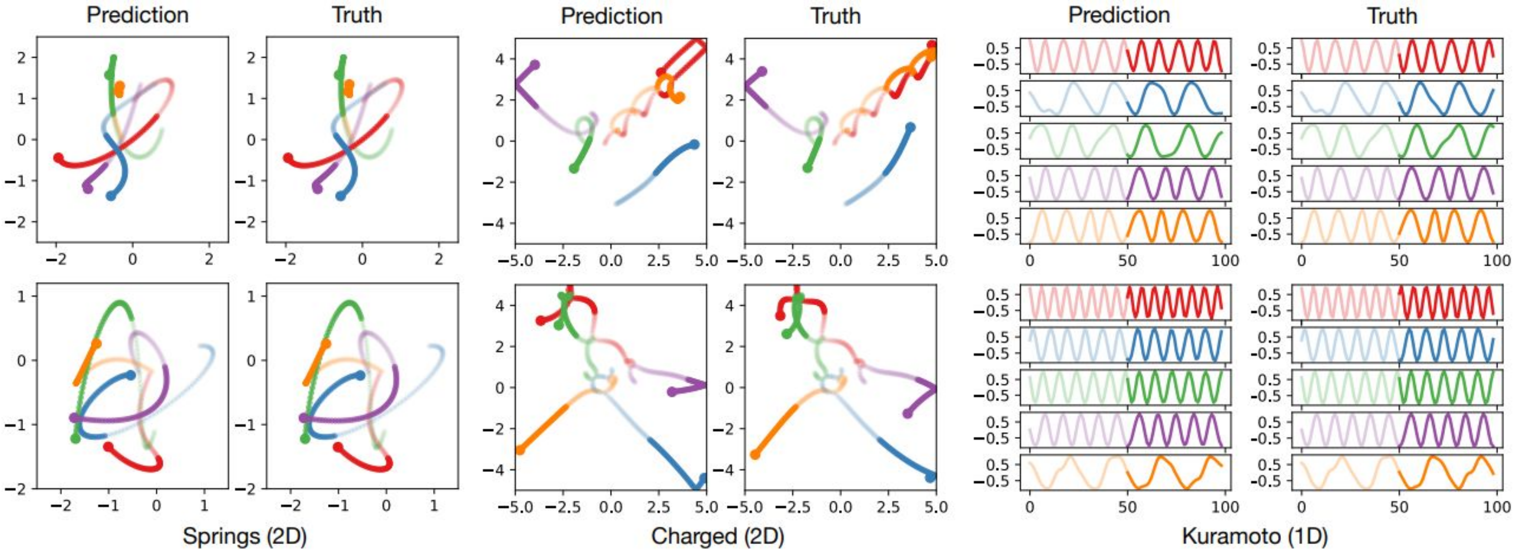


(a) Yahoo (b) Yelp

Yang, Z., Hu, Z., Salakhutdinov, R., & Berg-Kirkpatrick, T. (2017). Improved variational autoencoders for text modeling using dilated convolutions. *ICML 2017*

Recent successes: **Physics (interacting systems)**





Kipf, T., Fetaya, E., Wang, K. C., Welling, M., & Zemel, R. (2018). Neural relational inference for interacting systems. *ICML 2018*.

Conclusion

Generative modeling: the way to go to achieve AI.

Deep generative modeling: very successful in recent years in many domains.

GANs, VAEs, Autoregressive models and **Flow-based models**

Next steps: **video processing**, **better priors** and **decoders**, **geometric methods**, …

Conclusion

Generative modeling: the way to go to achieve AI.

Deep generative modeling: very successful in recent years in many domains.

GANs, VAEs, Autoregressive models and **Flow-based models**

Next steps: **video processing**, **better priors** and **decoders**, **geometric methods**, …

Conclusion

Generative modeling: the way to go to achieve AI.

Deep generative modeling: very successful in recent years in many domains.

GANs, VAEs, Autoregressive models and **Flow-based models**

Next steps: **video processing, better priors** and **decoders, geometric methods**, …

Conclusion

Generative modeling: the way to go to achieve AI.

Deep generative modeling: **very successful** in recent years in many domains.

GANs, **VAEs**, **Autoregressive models** and **Flow-based models**

Next steps: **video processing**, **better priors** and **decoders**, **geometric methods**, …

Conclusion

Generative modeling: the way to go to achieve AI.

Deep generative modeling: very successful in recent years in many domains.

GANs, **VAEs**, **Autoregressive models** and **Flow-based models**

Next steps: **video processing**, **better priors** and **decoders**, **geometric methods**, …

Code on github:

https://github.com/jmtomczak

Webpage:

http://jmtomczak.github.io/

Contact:

jmk.tomczak@gmail.com