Variational generation of images by deep spatial architectures

ARTHUR VIVÉ

MASTER’S PAPER PRESENTATION

ADVISOR: YALI AMIT

JULY 21 2017

Introduction

Context:

- Deep learning achieve State-of-the-Art in Computer Vision
- But little is known about the theory (optimization)

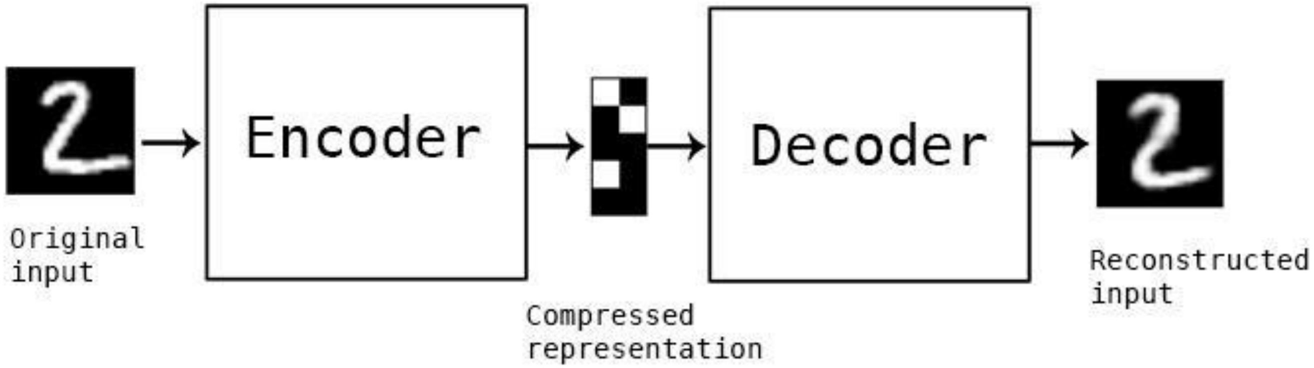
Goal:

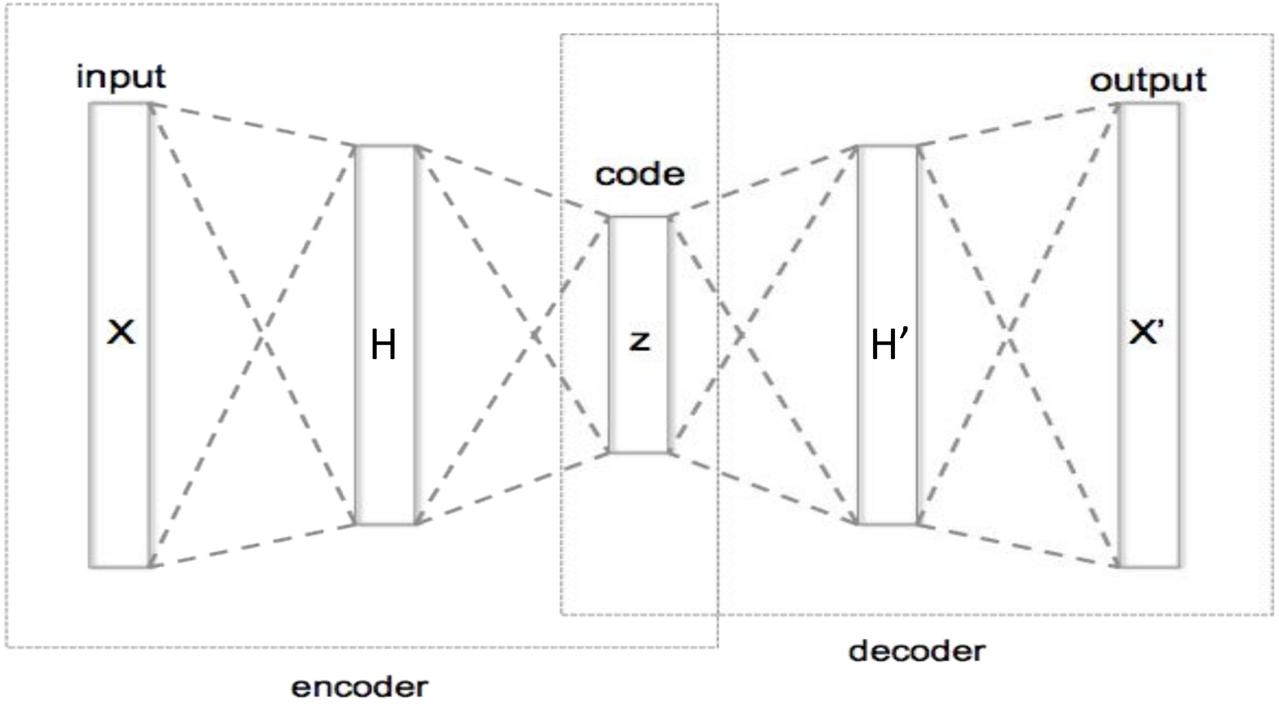
- Generate images
- Reconstruct (denoise)
- Use higher-level information

Cherry-picked sample from a Generative adversarial network (Goodfellow 2016)

Important Concepts

Deep Autoencoders





$$h = \sigma(W_1 x + b_1)$$
$$z = \sigma(W_2 h + b_2)$$
$$h' = \sigma(W_3 z + b_3)$$
$$x' = \sigma(W_4 h' + b_4)$$

Nonlinearity σ : sigmoid, tanh, max(0, .)

Loss: Cross-Entropy, Mean squared error

Properties:

- Gradient analytically tractable -> Training end to end.
- denoising / manifold learning, visualization.

Images: Keras.io and Wikipedia

Variational Inference

$$p(x,z) = p_\theta(x|z)p_\theta(z)$$

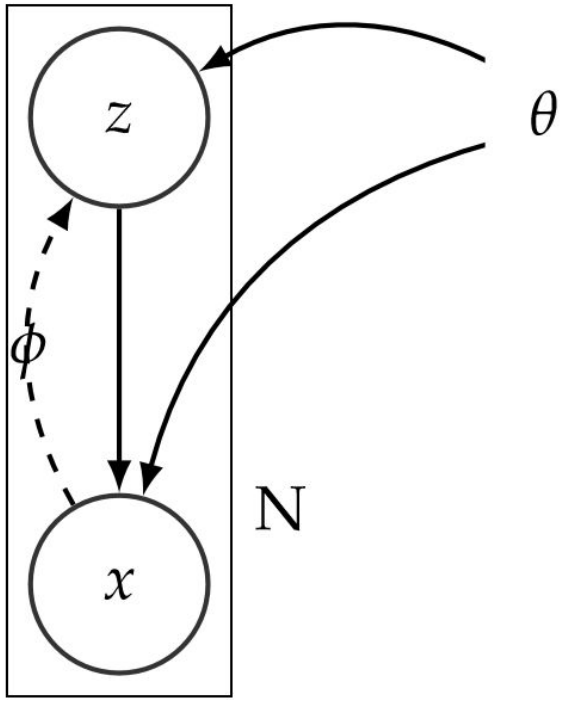


Figure 2: *Parametrization of the latent variable model.*

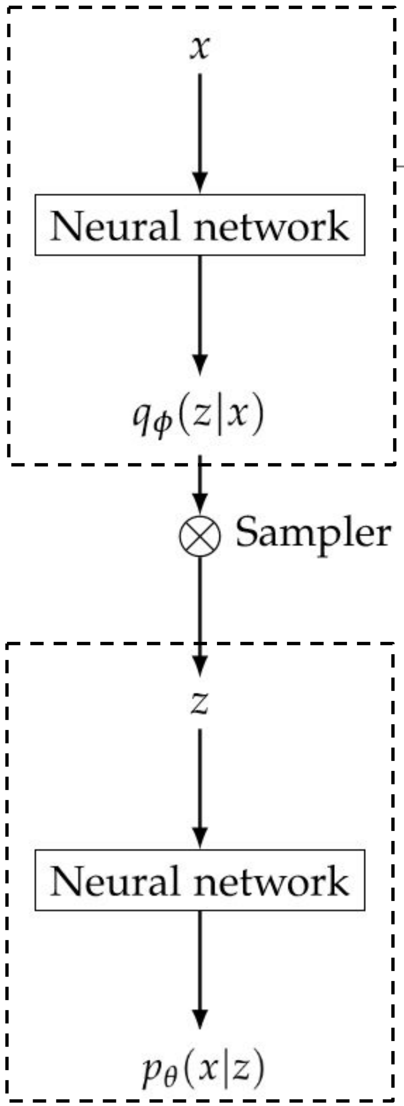
$$\begin{aligned} log[p(x)] &= -D_{KL}(q_\phi(z|x)||p_\theta(z)) + E_{z\sim q_\phi(.|x)}[log(p_\theta(x|z)] \\ &\quad + D_{KL}(q_\phi(z|x)||p_\theta(z|x)) \\ &\geq -D_{KL}(q_\phi(z|x)||p_\theta(z)) + E_{z\sim q_\phi(.|x)}[log(p_\theta(x|z)] \\ &=: L(x,\theta,\phi) \end{aligned}$$

True regardless of q, the approximate posterior.

- Maximize this tractable lower bound for an easily computable q.
- Fix the parametrization and optimize over (θ,φ)

Source: Michael I. Jordan, et al. An introduction to variational methods for graphical models. 1999.

Variational Autoencoder



Approximate posterior

$$\hat{L}(\theta, \phi, x) = -D_{KL}(q_\phi(z|x)||p_\theta(z)) + \frac{1}{L}\sum_{j=1}^{L} log p_\theta(x|z_j)$$

where (z_j) are sampled independently from $q_\phi(z|x)$.

Assumptions:

- Continuous latent variable z
- Differentiability of p and q w.r.t. θ, φ respectively.
- Necessary reparametrization:

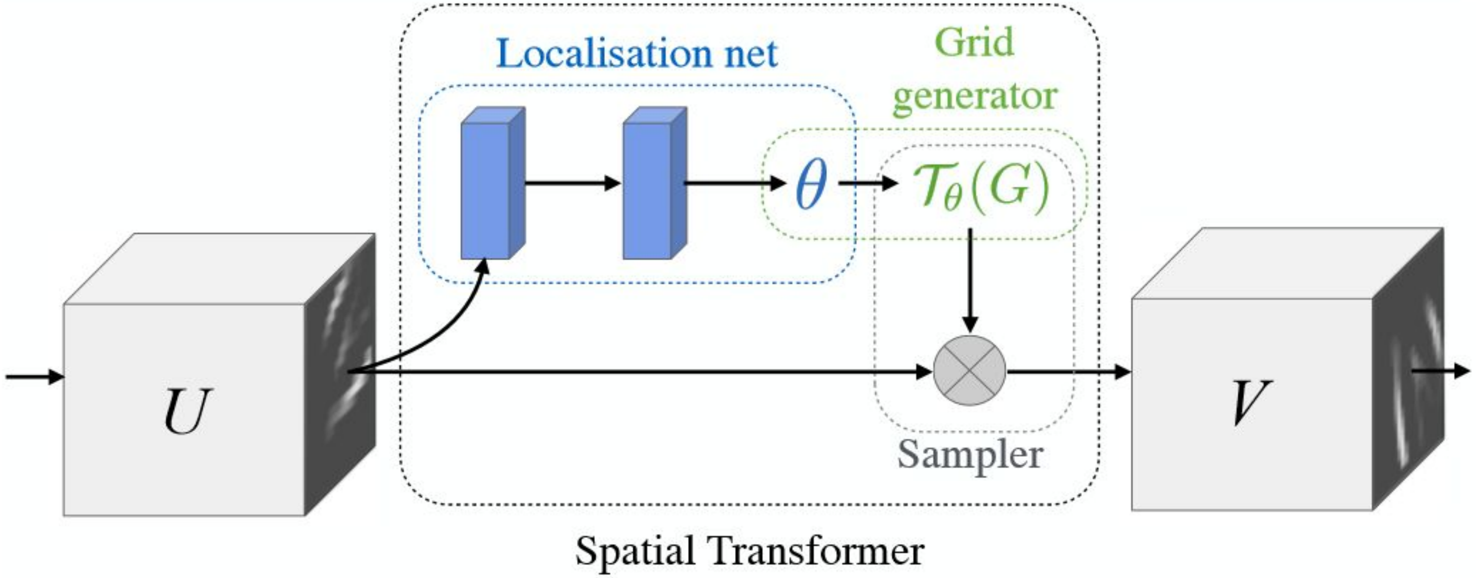
$$z_j = g_\phi(\epsilon_j, x), \text{ e.g. } z_j = \mu_\phi(x) + \sigma_\phi(x)\epsilon_j,\ \epsilon_j \sim N(0,1)$$

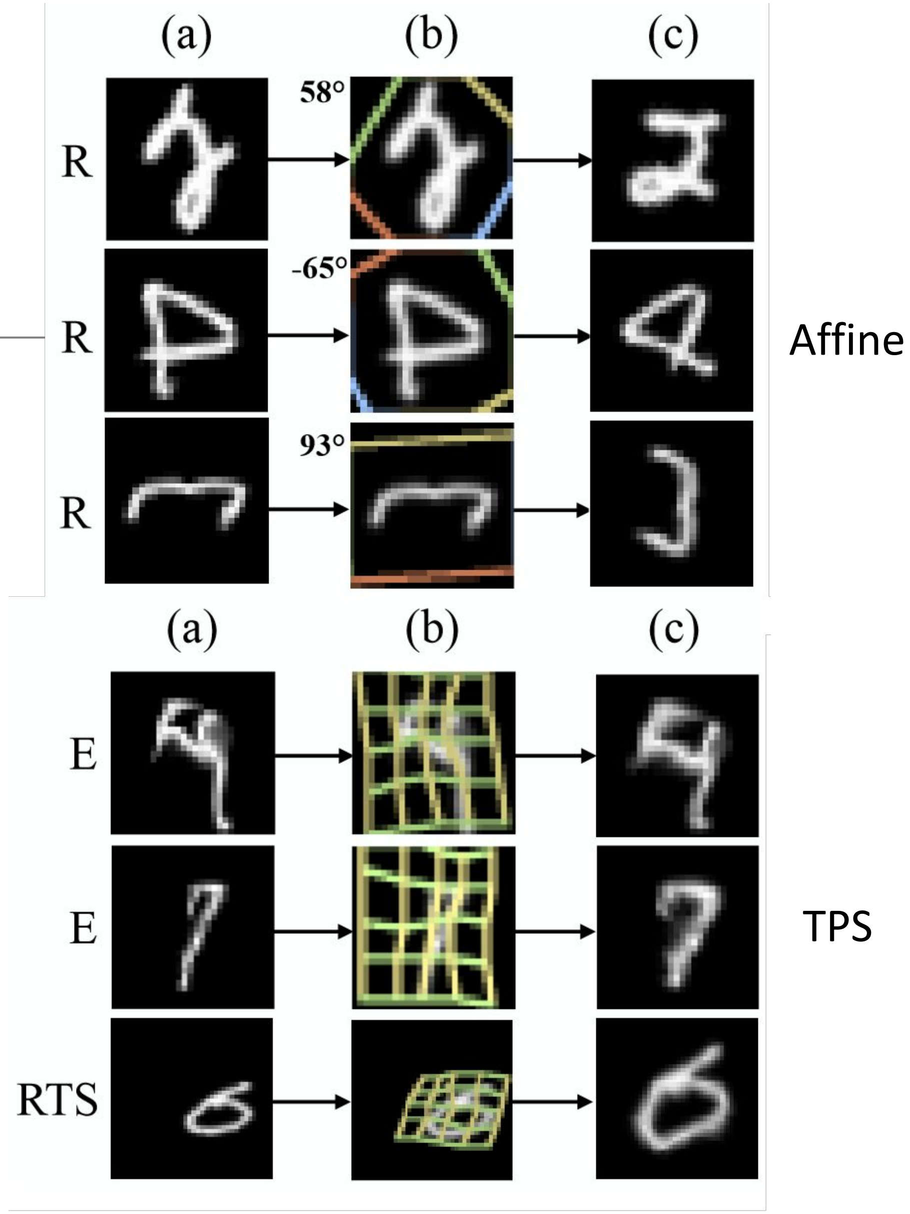
Generative model

Properties:

- Gradient analytically tractable -> Training end to end.
- denoising / manifold learning, visualization.
- Generative models

Spatial Transformer Layers





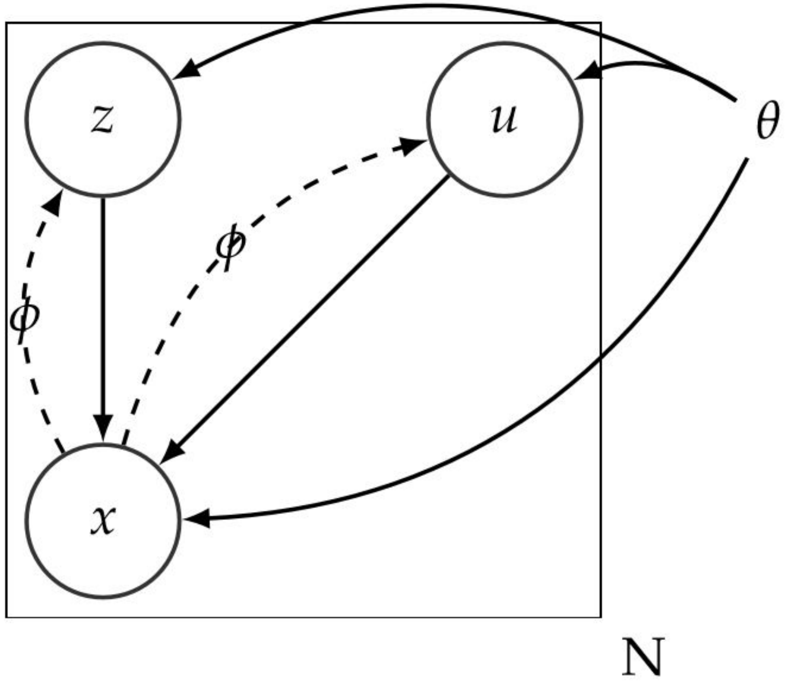
2 differentiable transformations allowed:

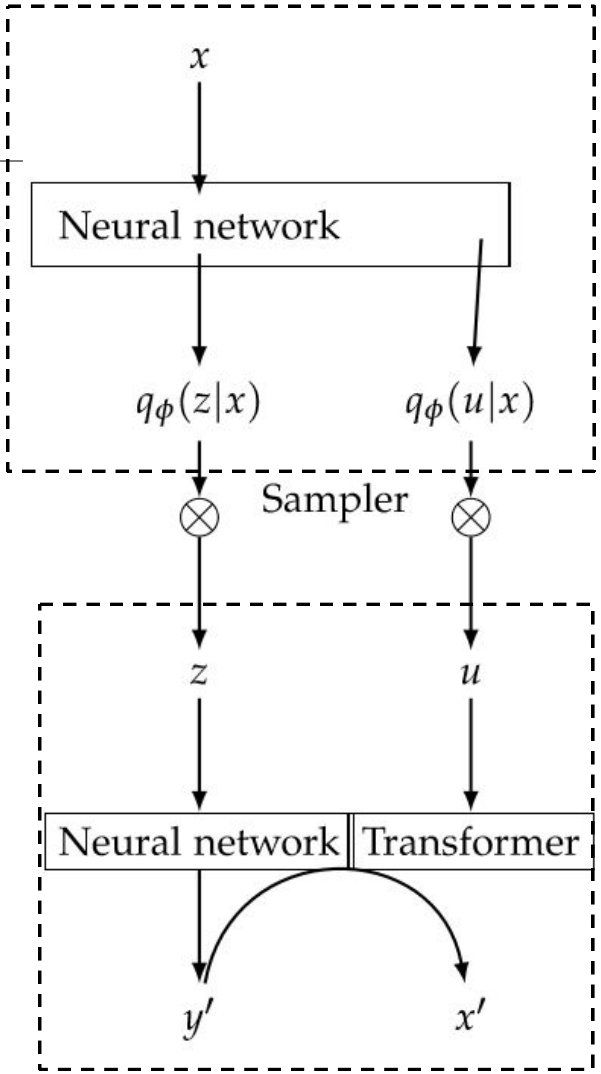
- Affine and thin plate splines

Learns the pose by end-to-end learning (no data about this pose)

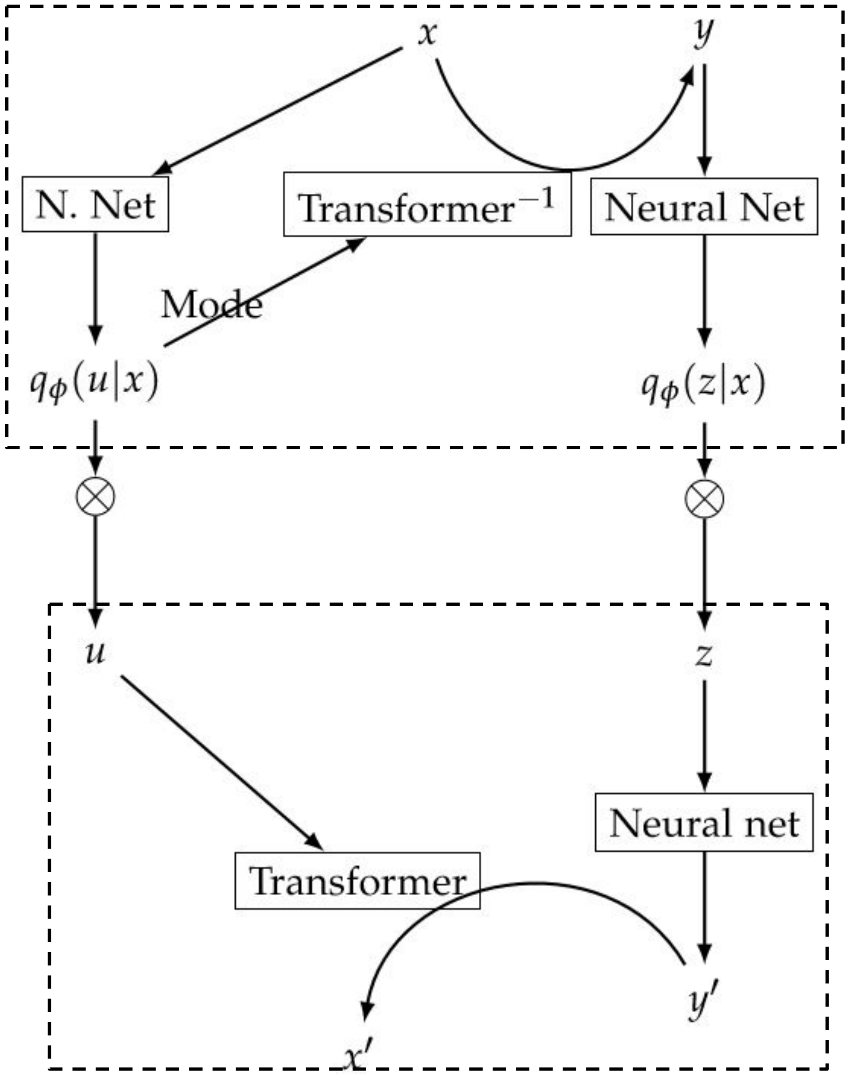
Our models

Our models: Transformer VAE and Double-Transformer VAE





TVAE



DTVAE

Dataset

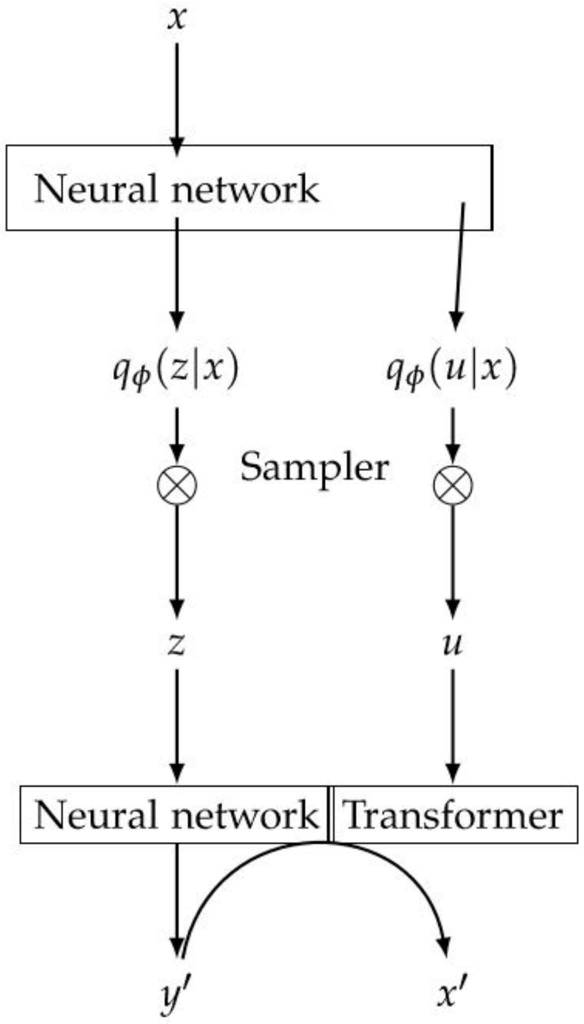
25 first digits of the MNIST test dataset

MNIST Dataset: 60,000 train digits between 0 and 9.
10,000 test samples

- Grayscale (between 0 and 1)
- 28x28 pixels (784-dimensional)

Widely used as benchmark.

Distributions



Priors:

$z \sim N(0, I)$

$u \sim ?$

Approximate posteriors:

$z|x \sim N(\mu^z(x), Diag(\sigma_1^z(x), ..., \sigma_D^z(x)))$

$u = \mu^u(x)$

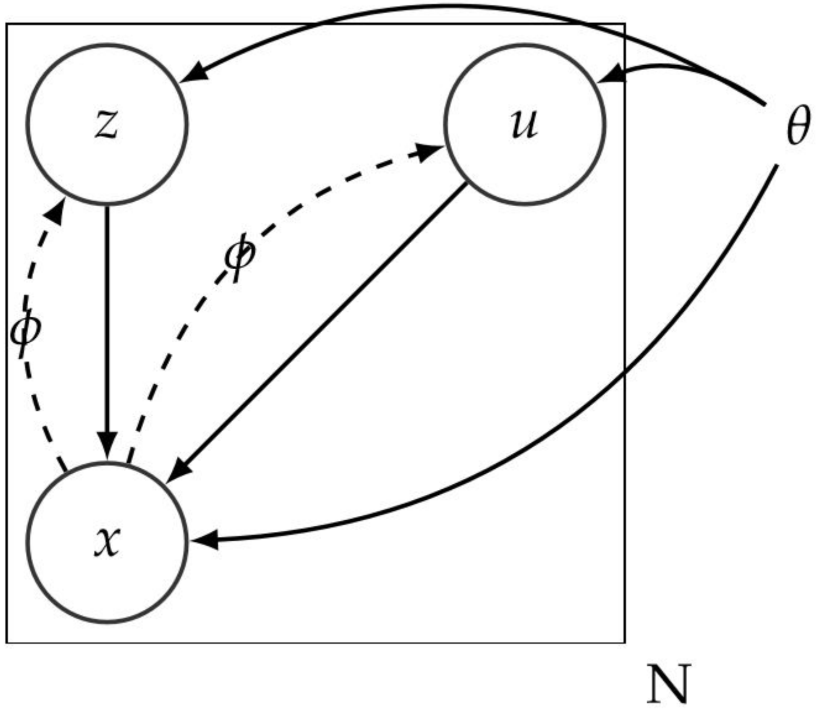
Conditional:

$x_{i,j} \sim Bernoulli(x'_{i,j})$

Functions of x are outputs of the encoder.

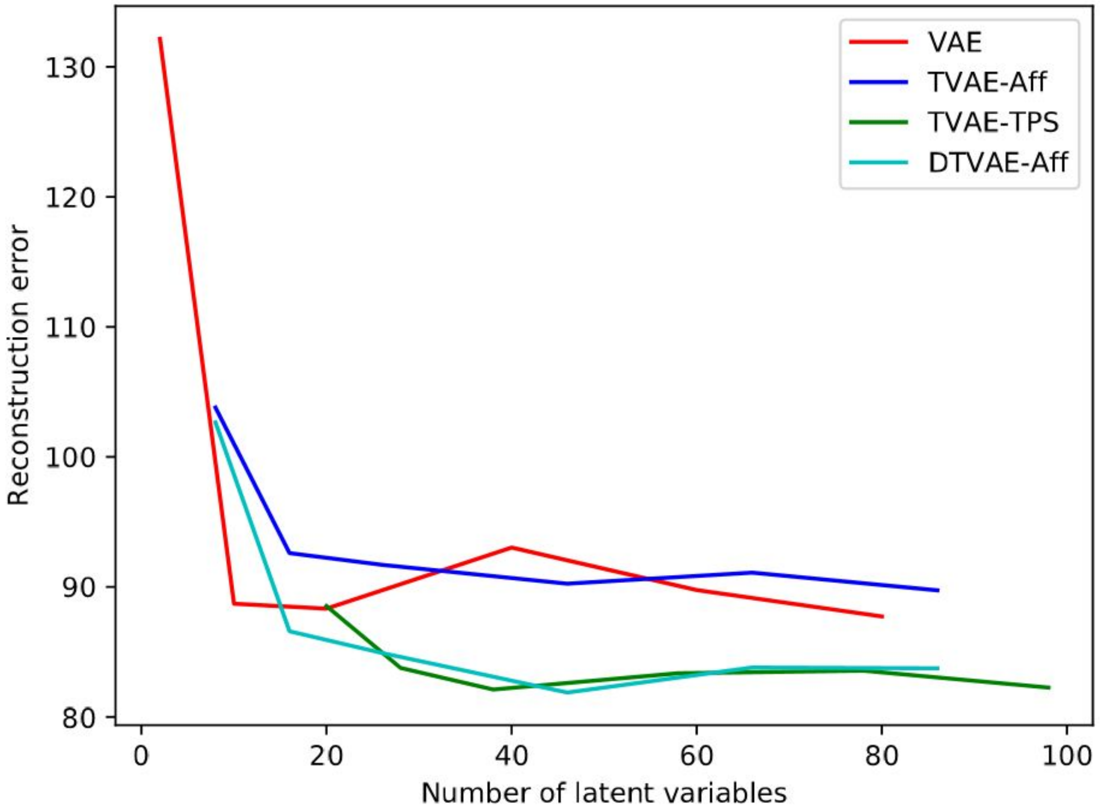
x' is the output of the decoder,

deterministic function of z and u.



- KL-Divergence has closed form; reconstruction error is a Cross-entropy (Binomial likelihood)

Reconstructing digits



Input

VAE

Aff - TVAE

TPS- TVAE

Aff- DTVAE

Linear separation in the latent space

Input	Mean Accuracy (%)	Standard deviation (%)	Test accuracy
Raw MNIST ($28 \times 28 = 784$)	91.19	0.74	91.80
PCA (8)	75.09	1.32	75.51
VAE (8)	87.70	0.93	87.99
Affine-TVAE (8)	89.30	0.68	88.39
TPS-TVAE (8)	88.42	0.58	88.65
Affine-DTVAE (8)	**93.56**	0.64	**93.87**
PCA (20)	85.62	1.20	86.52
VAE (20)	85.77	1.22	86.45
Affine-TVAE (20)	90.80	0.66	90.51
TPS-TVAE (20)	**94.65**	0.46	**94.48**
Affine-DTVAE (20)	**93.56**	0.61	**94.17**

Table 1: *Accuracy of linear SVMs trained on latent codes produced by different models on MNIST (higher is better). Dimension of the input z in parenthesis. To compute the mean accuracy and the standard deviation, we cross validated the models on the training set 9 times.*

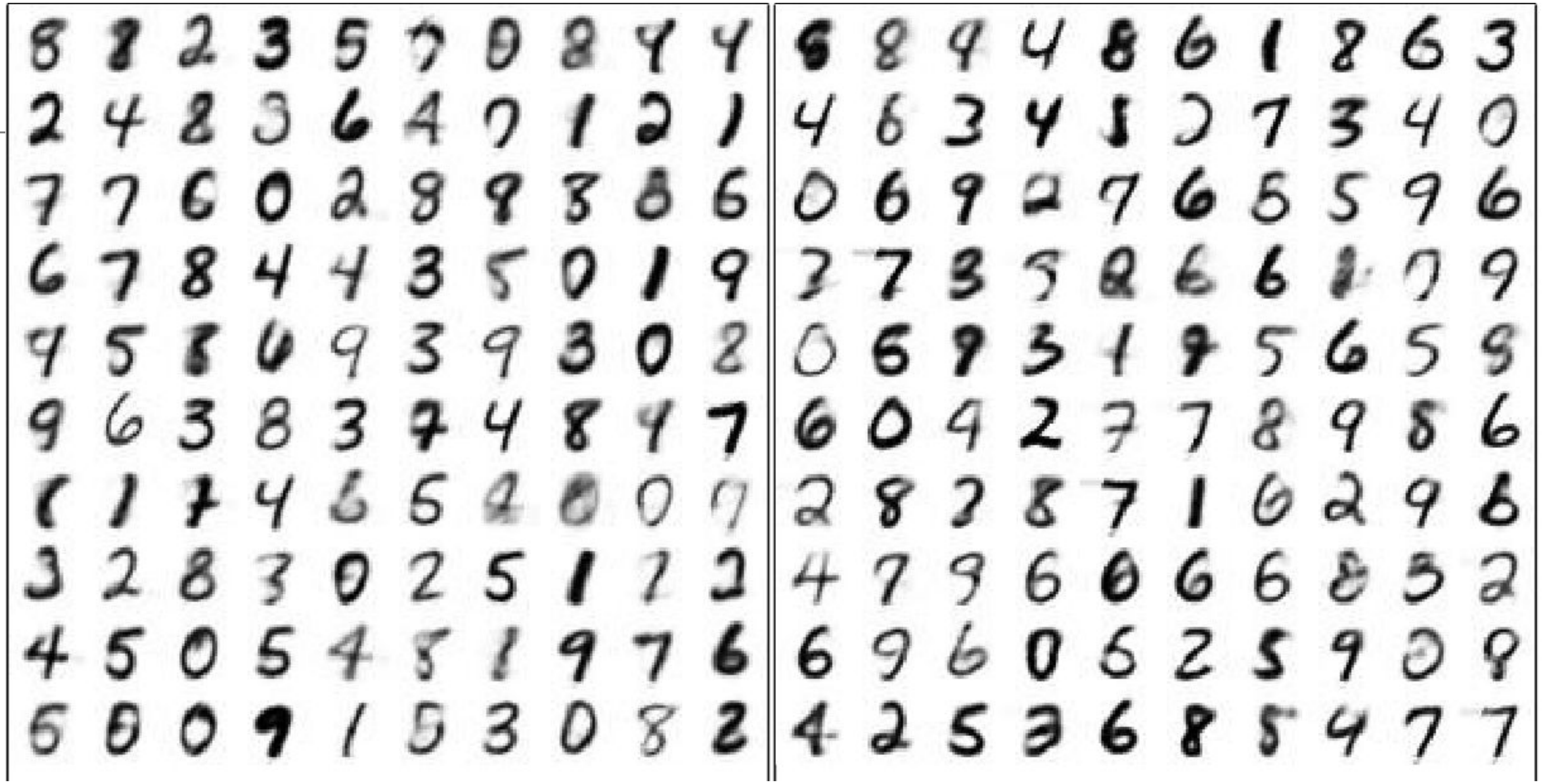
Generating Digits

VAE

Affine - TVAE

10-dimensional latent code (=dim u + dim z)

Generating Digits

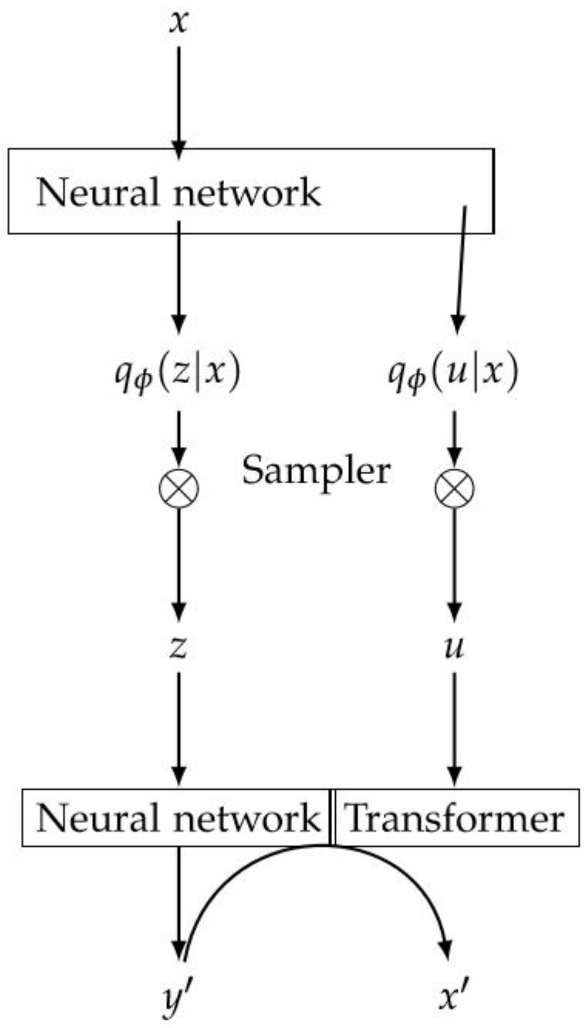


Affine - TVAE

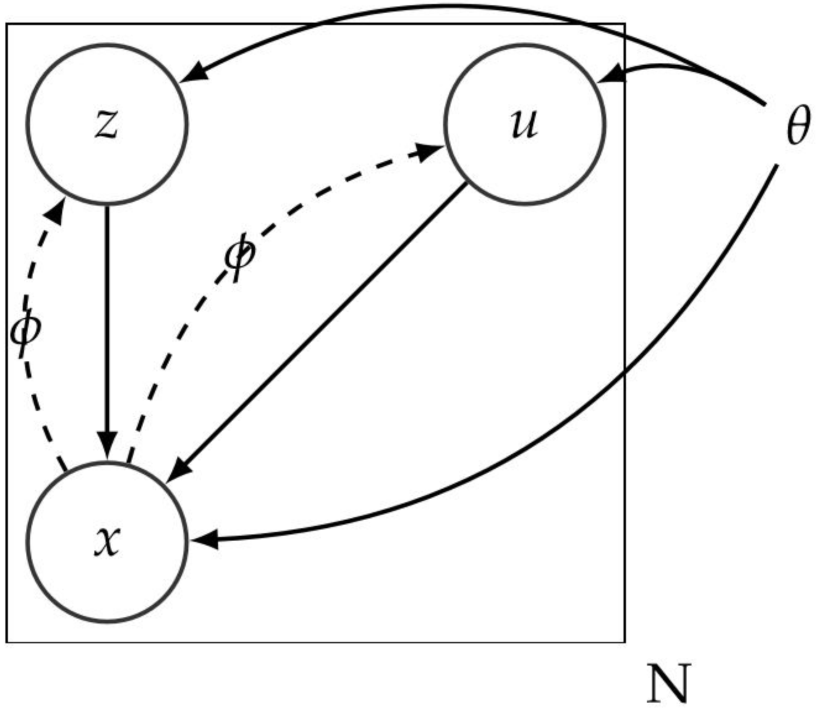
Affine DTVAE

10-dimensional latent code (=dim u + dim z)

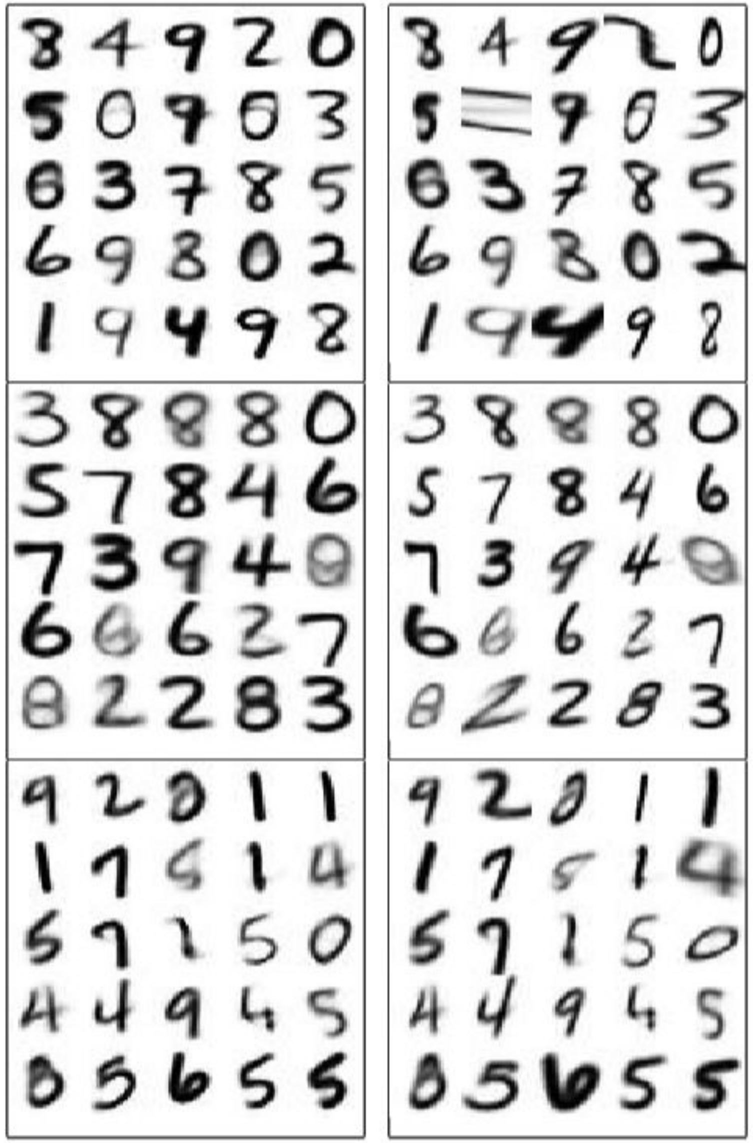
Prior specification on spatial variables



Priors:
$z \sim N(0, I)$
$u' \sim N(0, I)$
Approximate posteriors:
$z|x \sim N(\mu^z(x), Diag(\sigma_1^z(x), ..., \sigma_D^z(x))$
$u'|x \sim N(\mu^u(x), Diag(\sigma_1^u(x), ..., \sigma_S^u(x))$
Conditional:
$x_{i,j} \sim Bernoulli(x'_{i,j})$
Functions of x are outputs of the encoder.
x' is the output of the decoder
deterministic function of z and u' (through u).
With $u = Wu' + b$
Hence $u \sim N(b, WW^T)$
we estimate the prior parameters



Generating Digits II



Reference pose - Sampled pose

Gaussian priors:

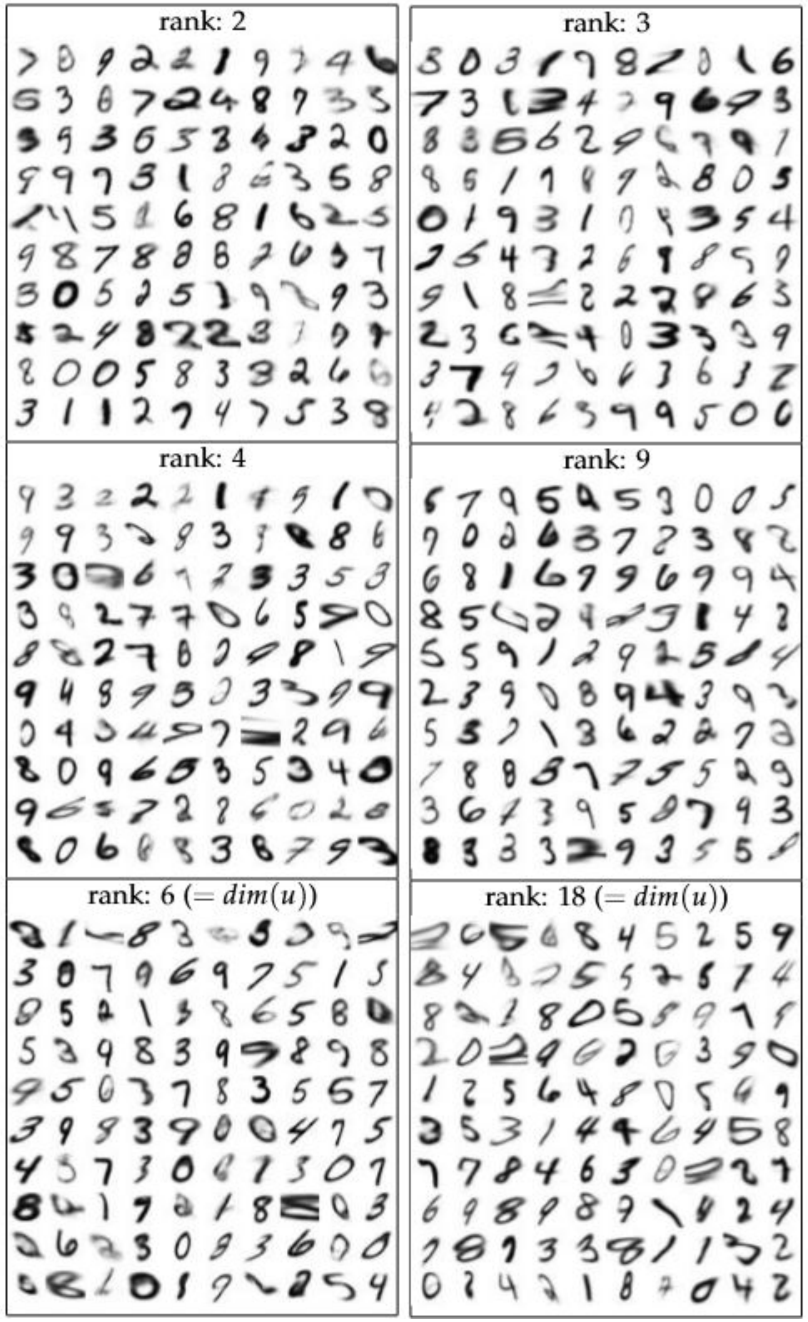
Centered on identity. Estimated diagonal covariance

Estimated mean and diagonal covariance

Estimated mean and diagonal covariance on the reparametrization (6 d.o.f)

Generating Digits III

Prior: estimated mean and covariance.
Rank of the covariance constrained by dim(u')



Affine-TVAE TPS-TVAE

Pros and cons

- Better Upright digits
- Better linear separation.
- Sharper than VAE
- Statistically relevant

- More complicated model
- Spatial prior difficult to get right
- Lack of objective measure of good generation
- Lacks discreteness

Future work

- Other datasets
- Add discreteness
- Better reconstruction error

Acknowledgment

Thank you for your attention
Questions?

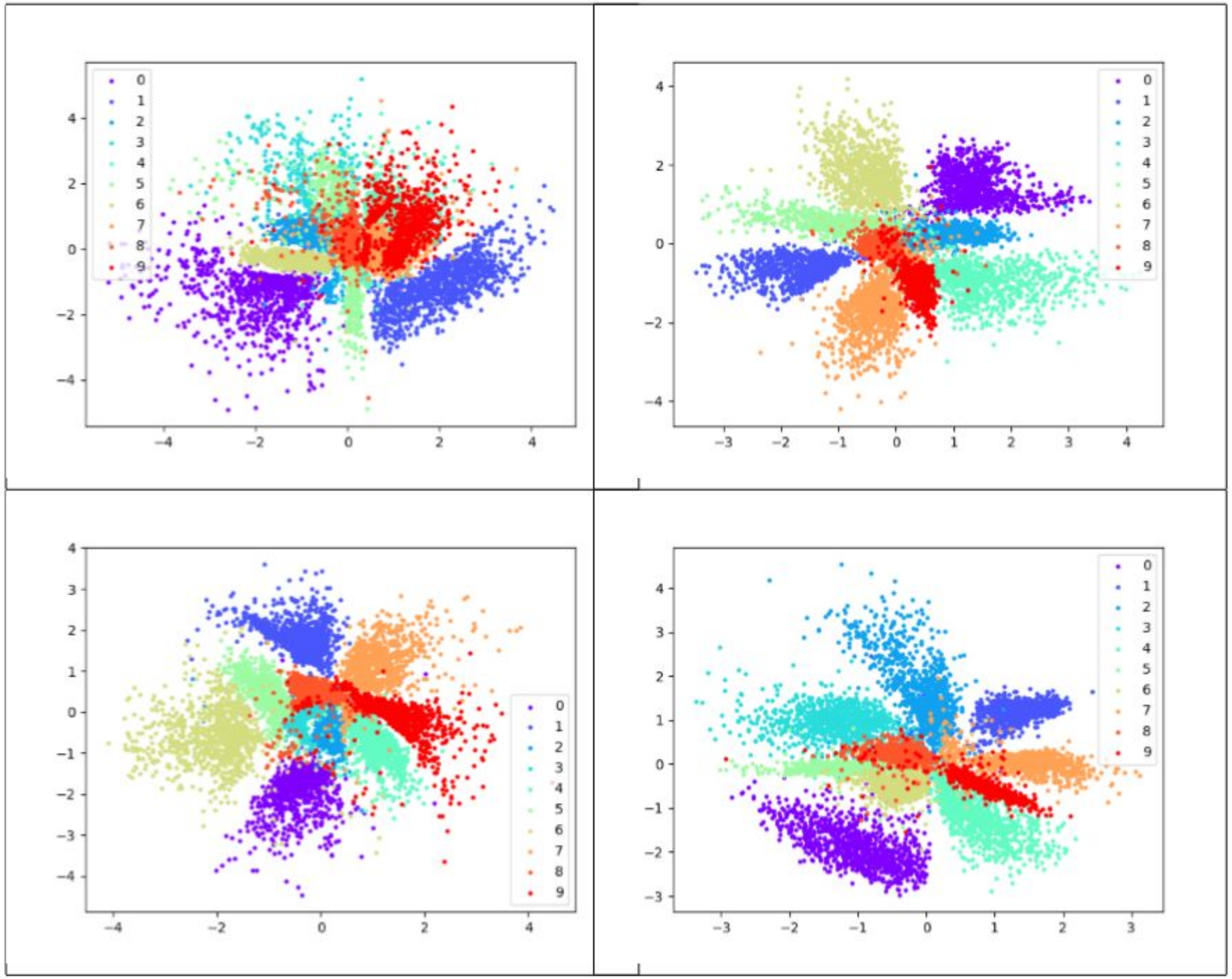


Figure 8: *Validation samples in a 2d-latent space.Top left: VAE.Top right: Affine TVAE, Top right: Affine TVAE. Bottom left: Affine DTVAE. Bottom right: TPS TVAE.*

Local optimum

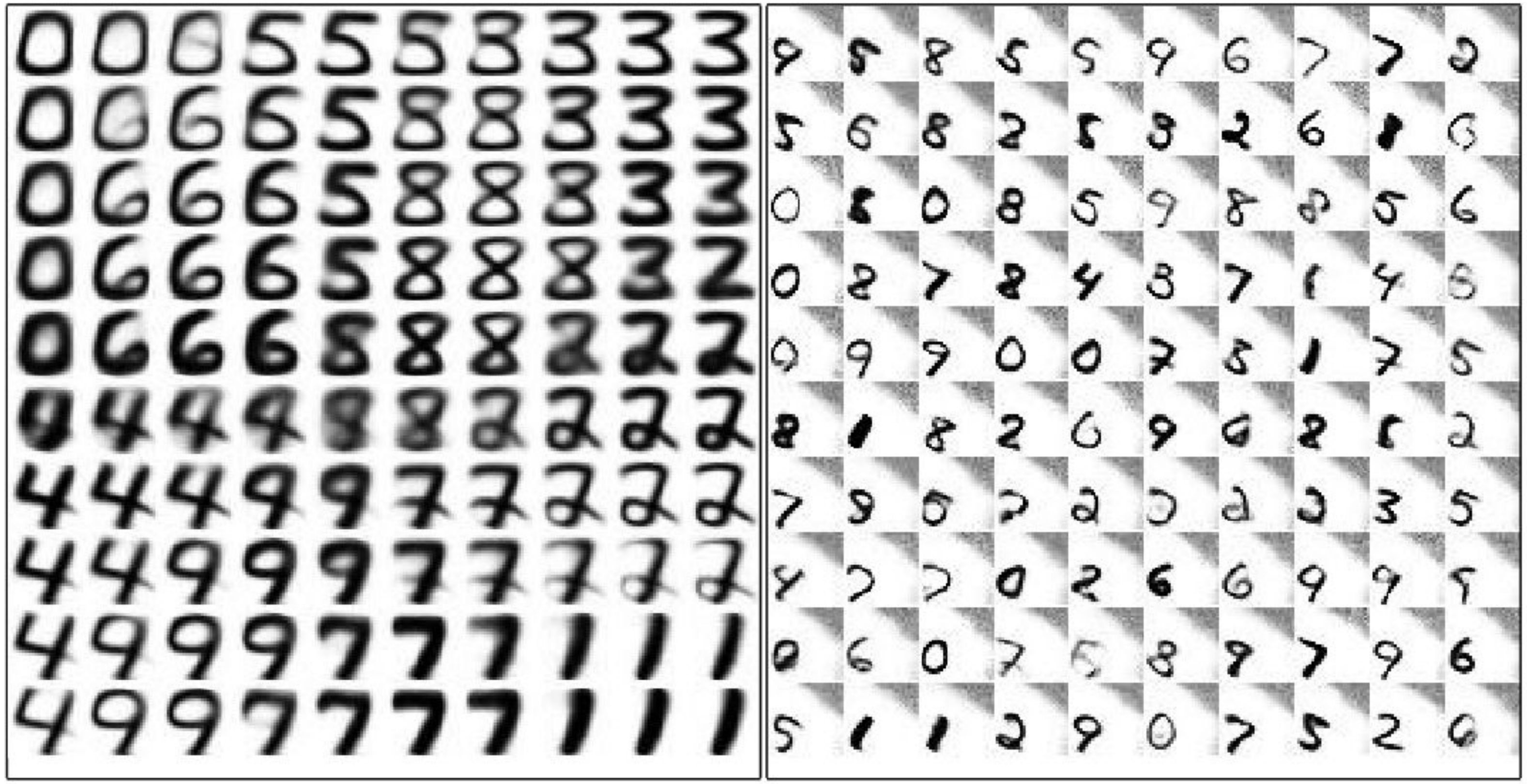


Figure 11: *Different outputs of the decoder with transformation set to identity. Left: example a promising local minimum. Left: example of a poor local minimum.*

Other examples of generation