Deep Learning

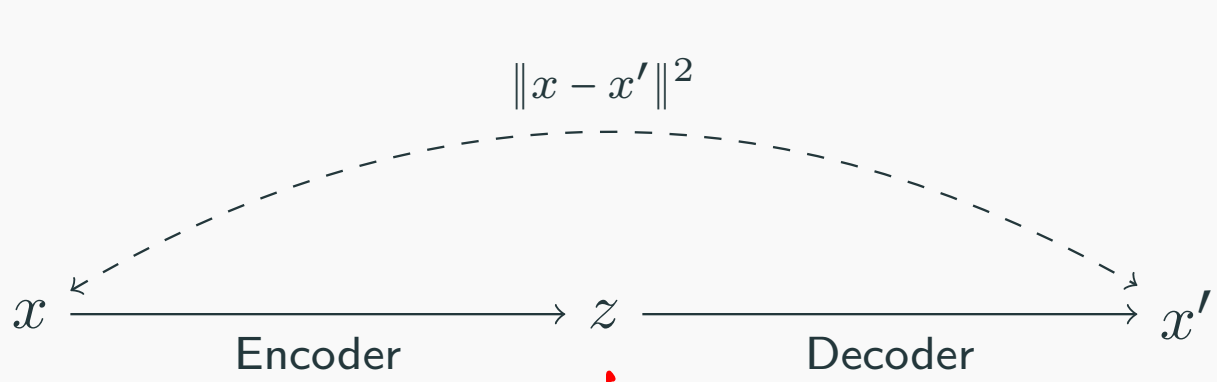
Ulf Brefeld & Soham Majumder

build: June 11, 2024

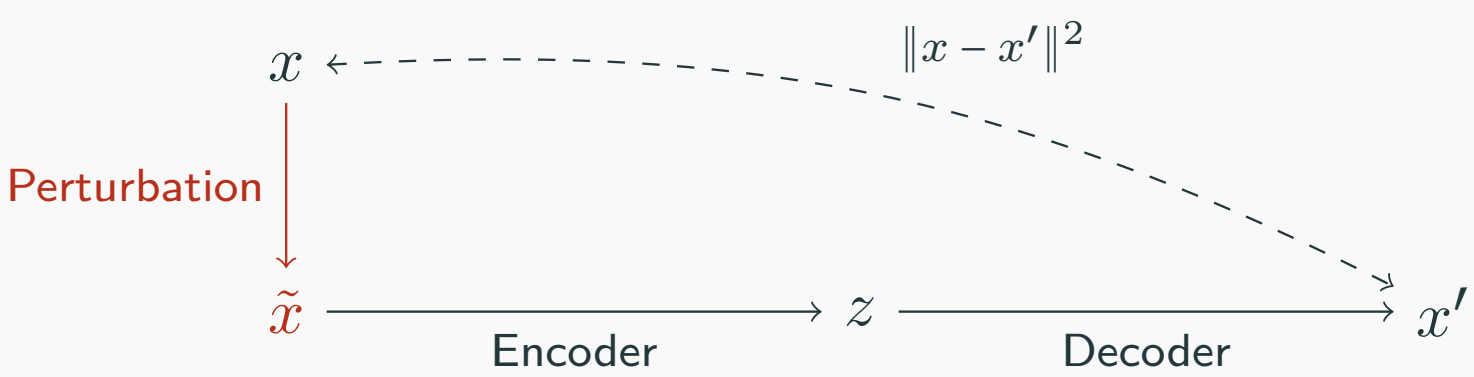
Machine Learning Group
Leuphana University of Lüneburg

Denoising Autoencoders

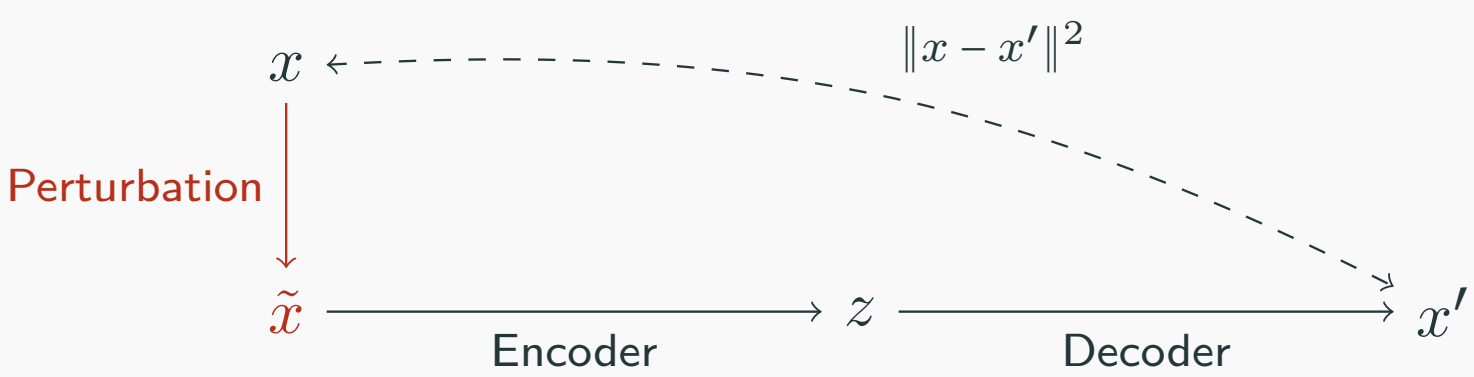
Autoencoders provide a framework for self-supervised learning.
Their goal is to reconstruct inputs x from latent codes z:



This can be extended to other tasks by attempting to reconstruct x from a perturbed version $\tilde{x}$:

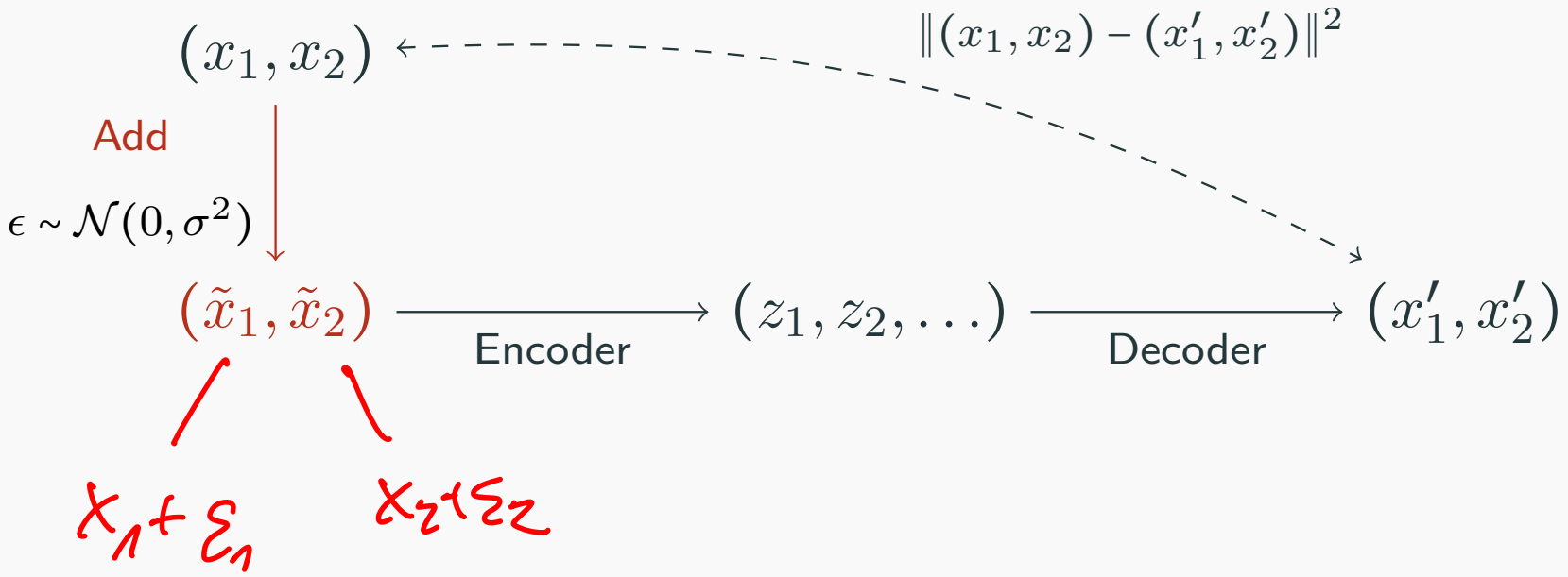


This can be extended to other tasks by attempting to reconstruct x from a perturbed version $\tilde{x}$:



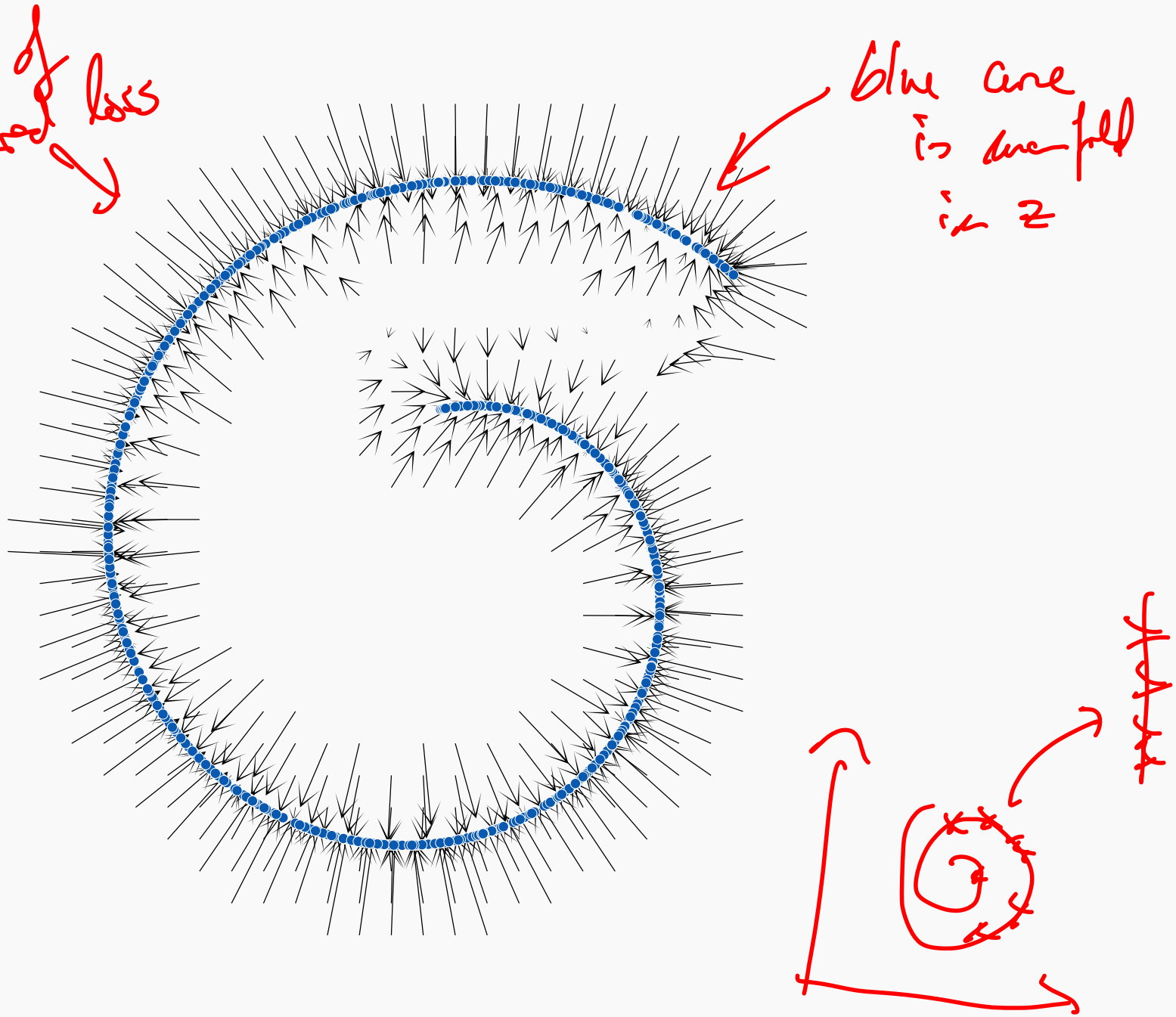
Intuitively, the perturbations encourage the model to learn more about the data, since correct reconstructions should be made even when some of the information is corrupted.

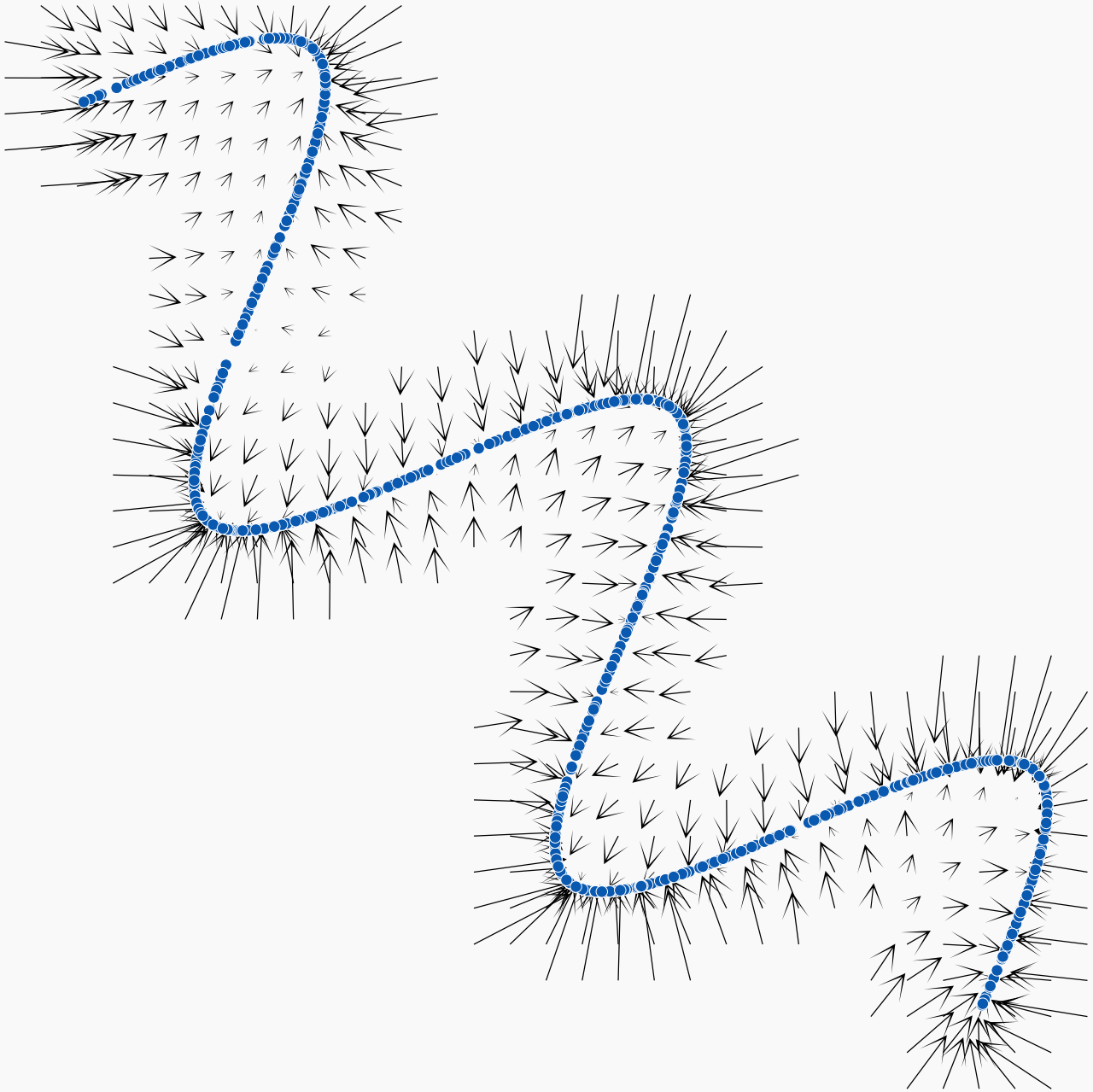
For an illustration of this intuition, consider an autoencoder for a two-dimensional problem. Suppose we corrupt the inputs by adding noise to them. In this case, the noise is a 2d vector sampled from a Gaussian distribution with small variance σ^2.

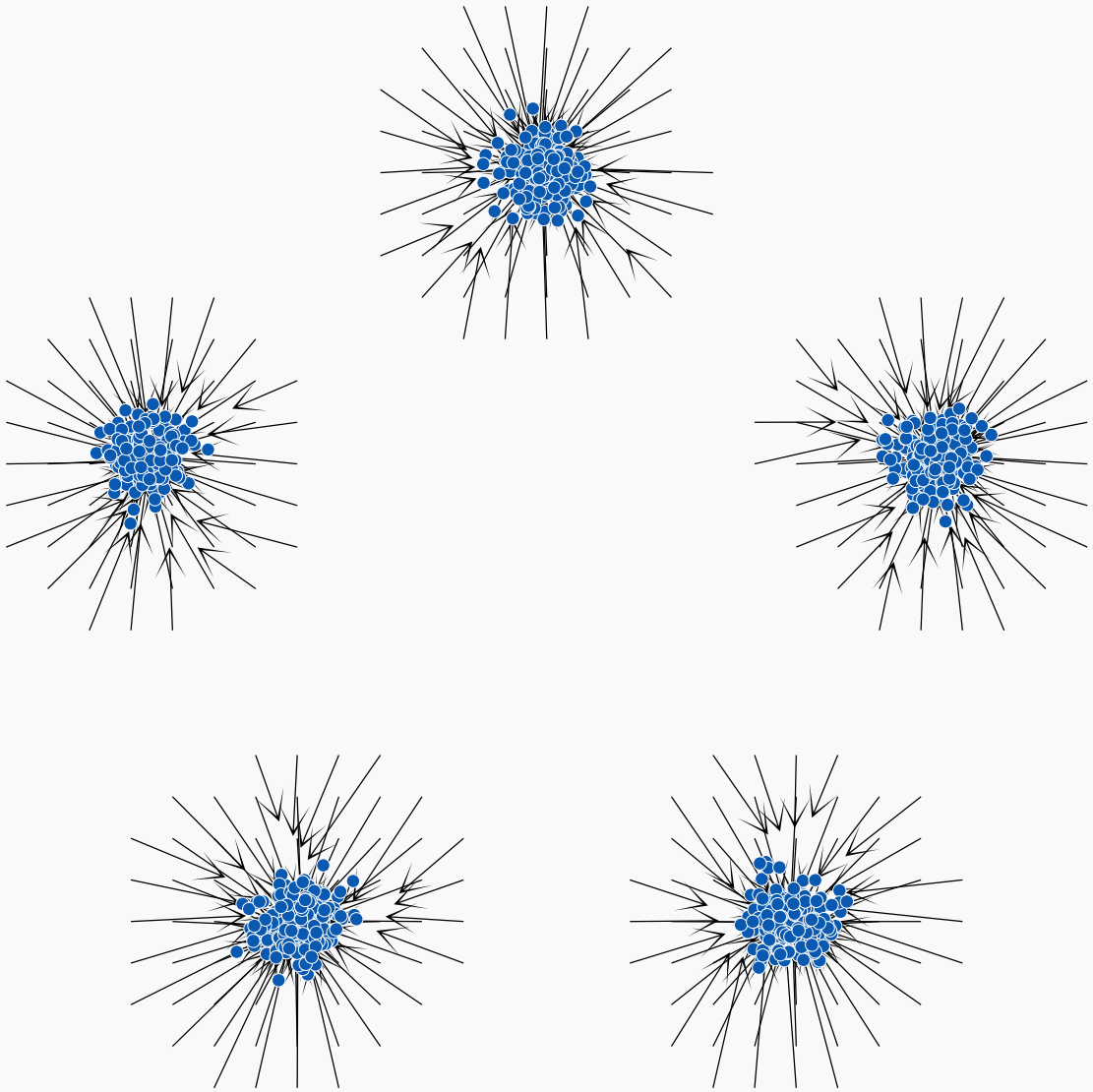


In the following, we inspect the correction introduced by the denoising autoencoder described earlier.

The arrows represent the movement of inputs $\tilde{x}$ into reconstructions x'. The inputs are sampled from a grid around the training data.


how of
speed loss
blue arc
is manifold
in z



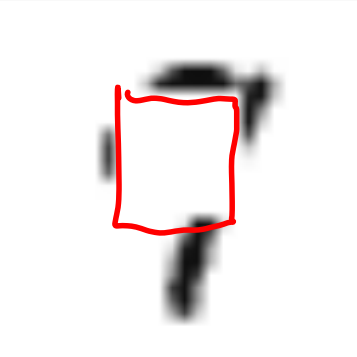


Naturally, images can also be corrupted and then an autoencoder is trained to restore the original. In this case, we can consider ways of corrupting the inputs which are specific to images:

Original

Pixel erasure

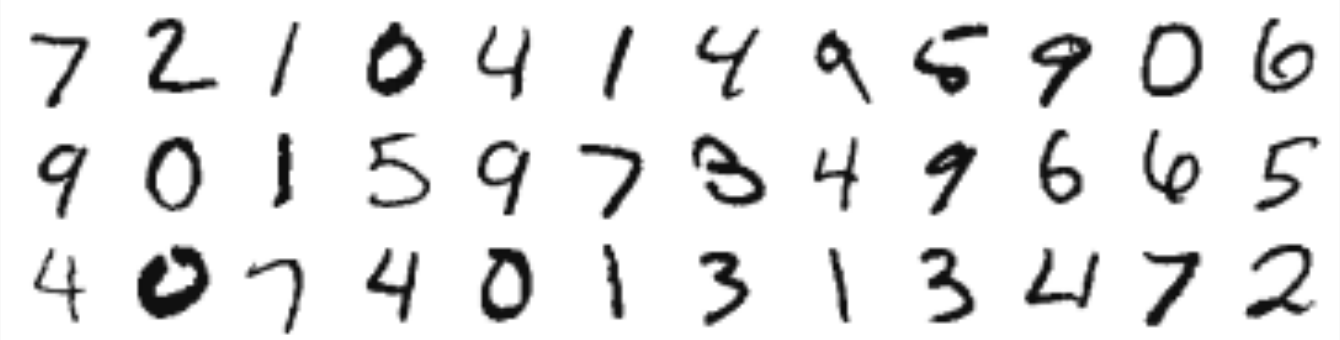
Blurring



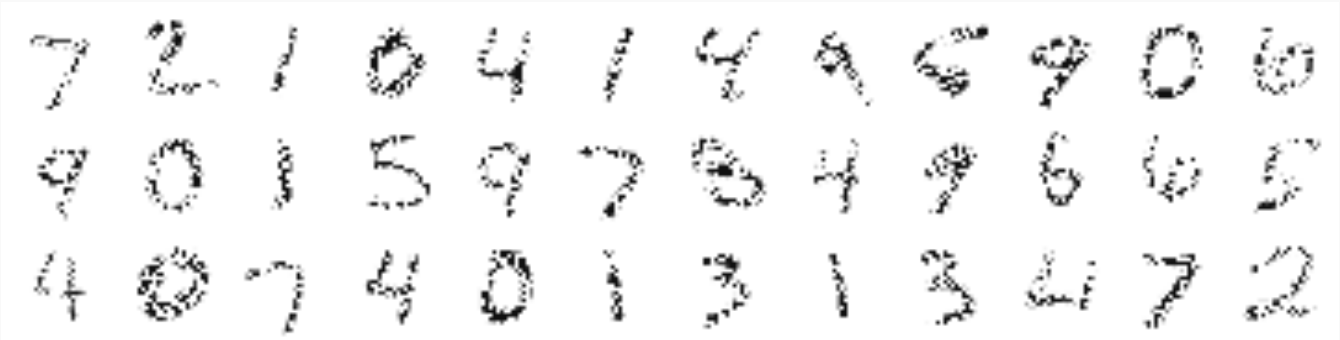
Block masking

With the autoencoder from the previous lecture, one could achieve the following results.

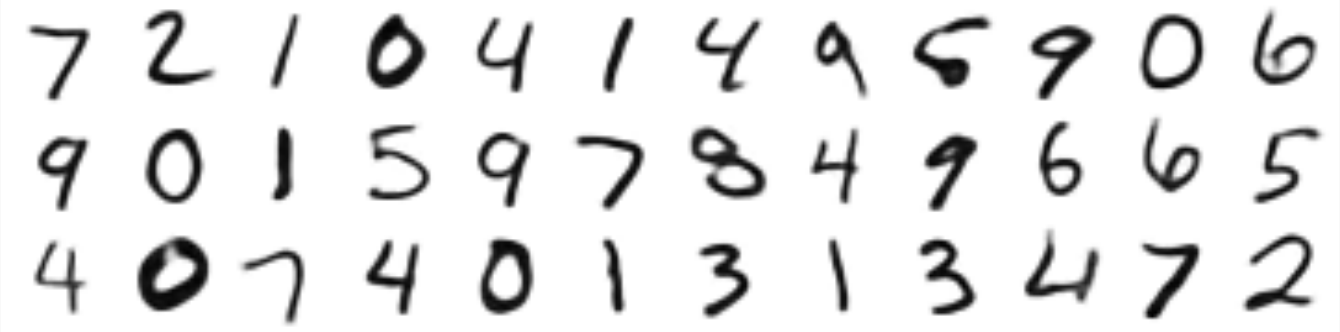
Original



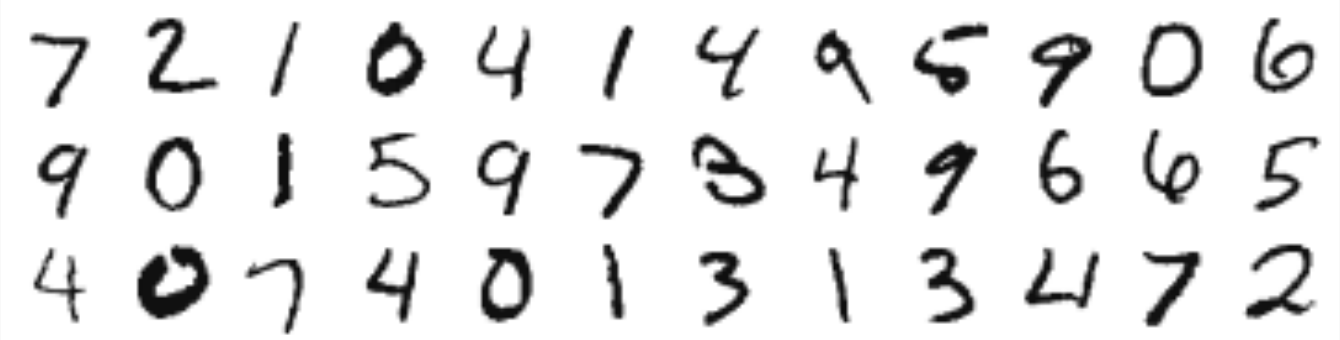
Corrupted ($p = 0.5$)



Reconstructed

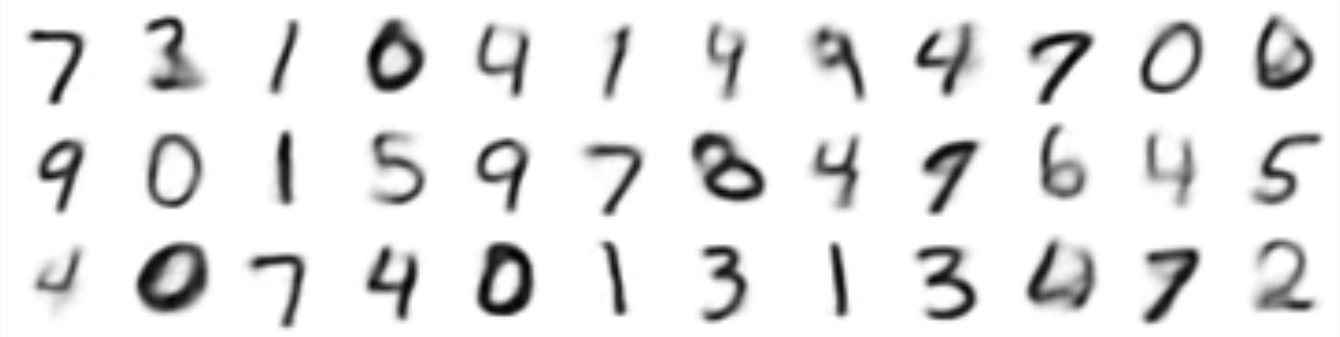


Original

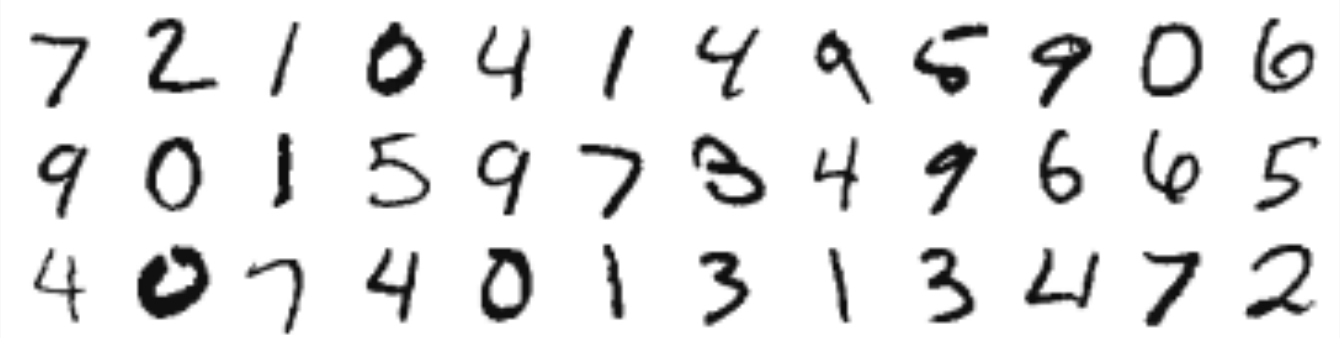


Corrupted ($p = 0.9$)

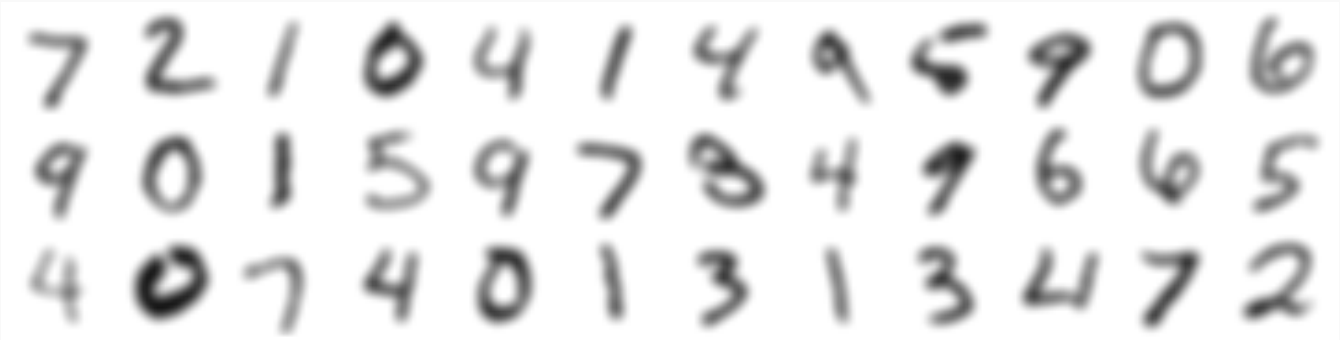
Reconstructed



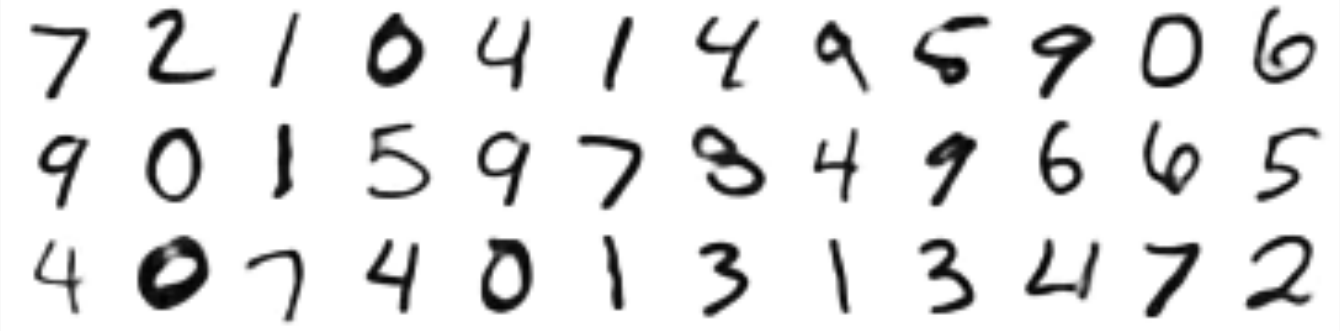
Original



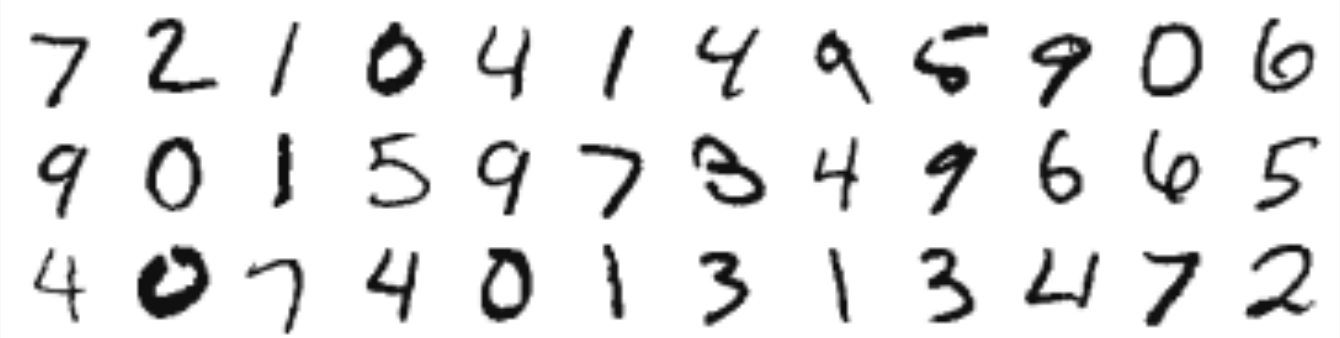
Corrupted ($\sigma = 2$)



Reconstructed

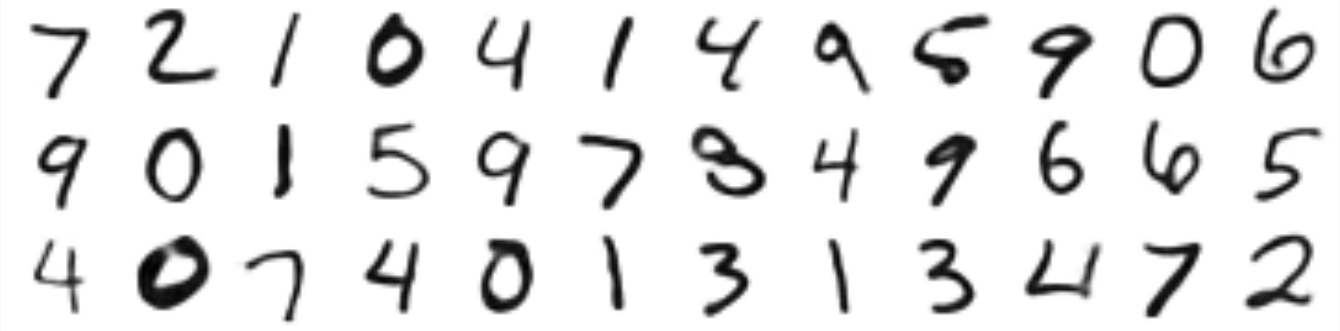


Original

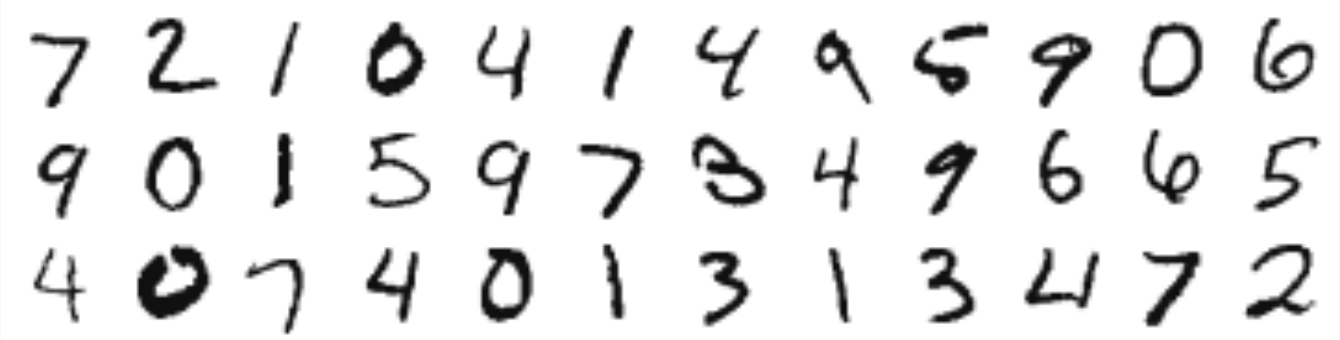


Corrupted ($\sigma = 4$)

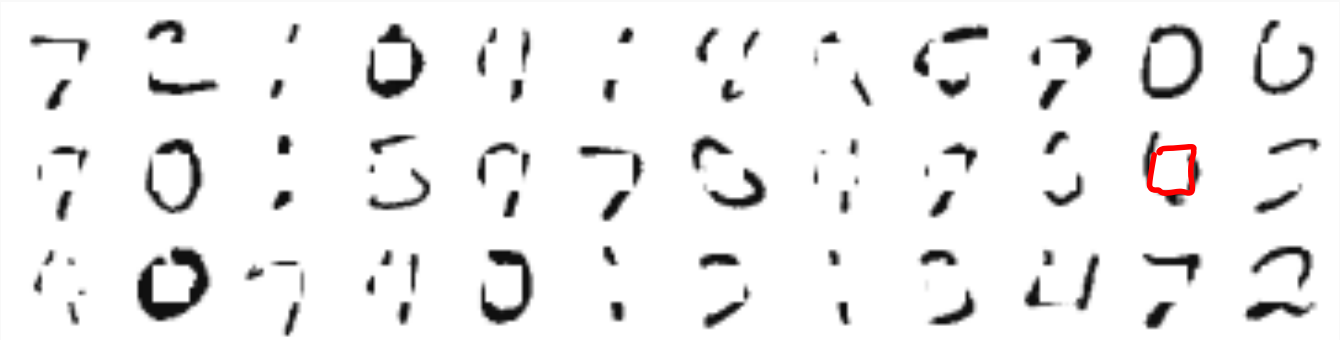
Reconstructed



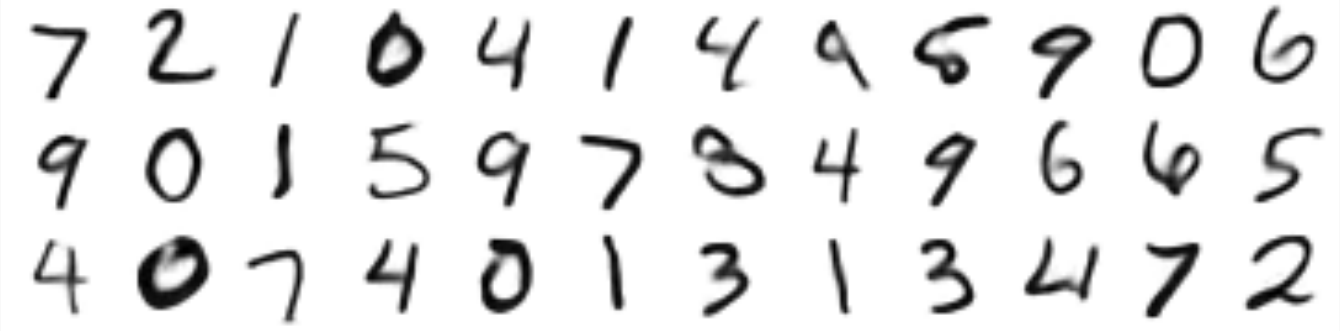
Original



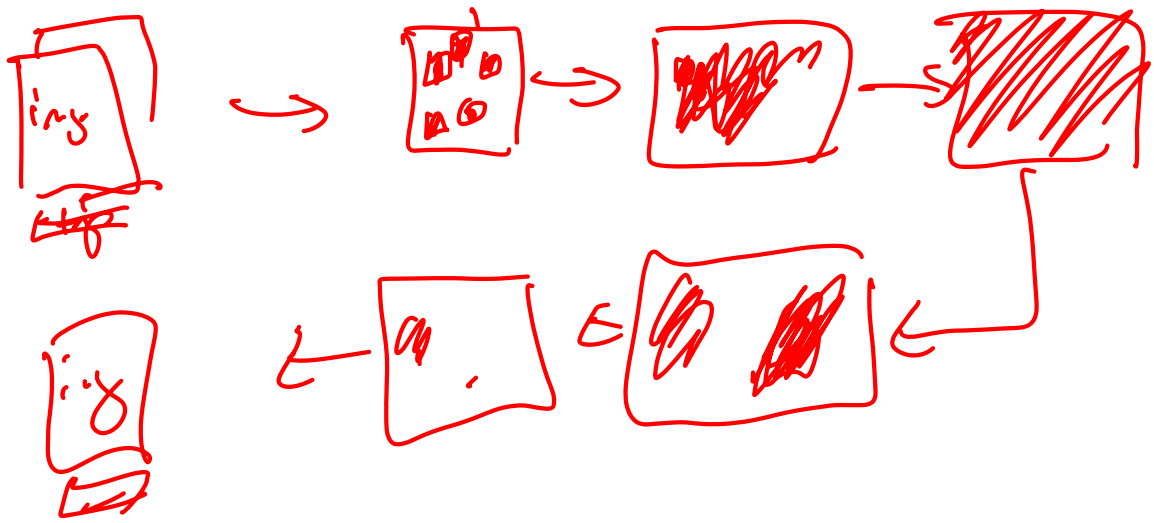
Corrupted (10×10)



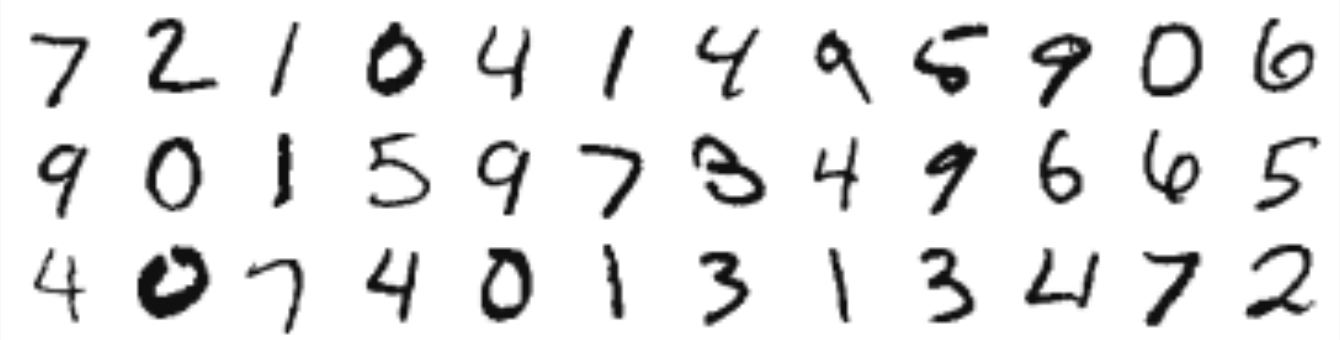
Reconstructed



Eg: Diffusion process

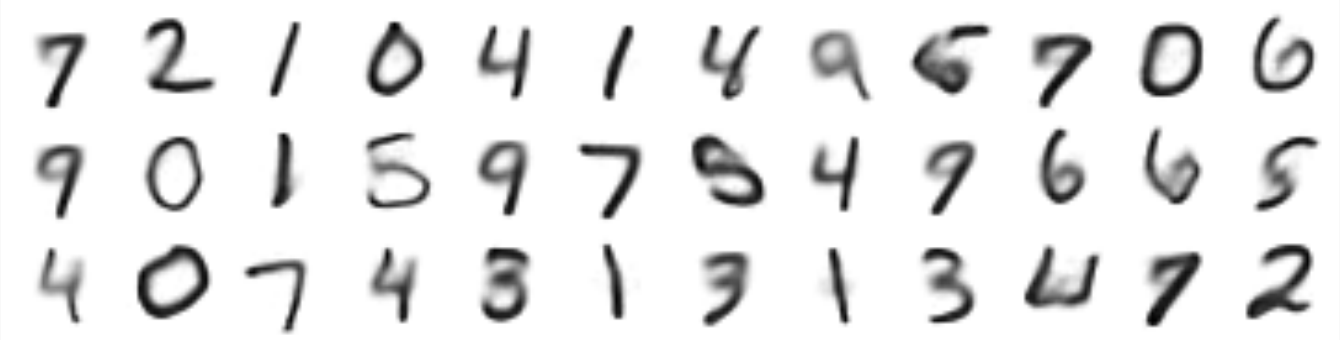


Original



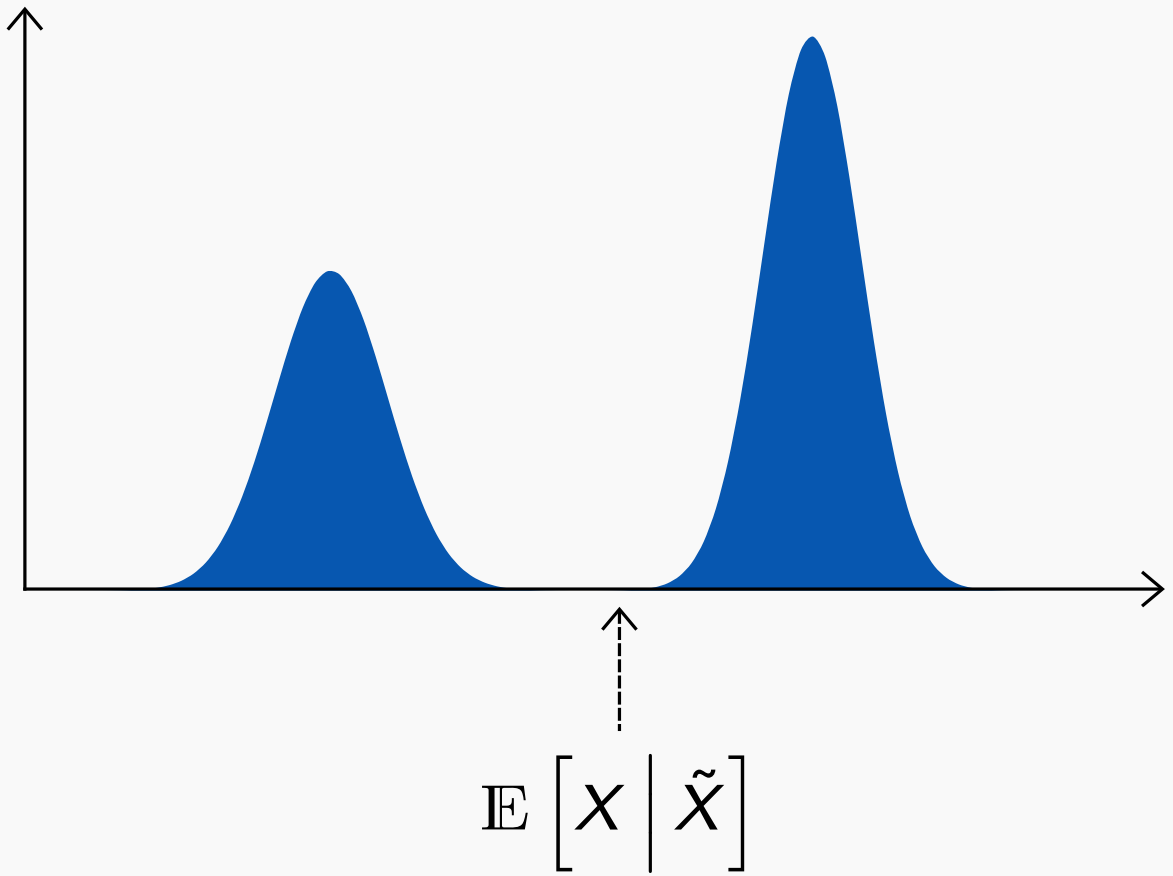
Corrupted (16×16)

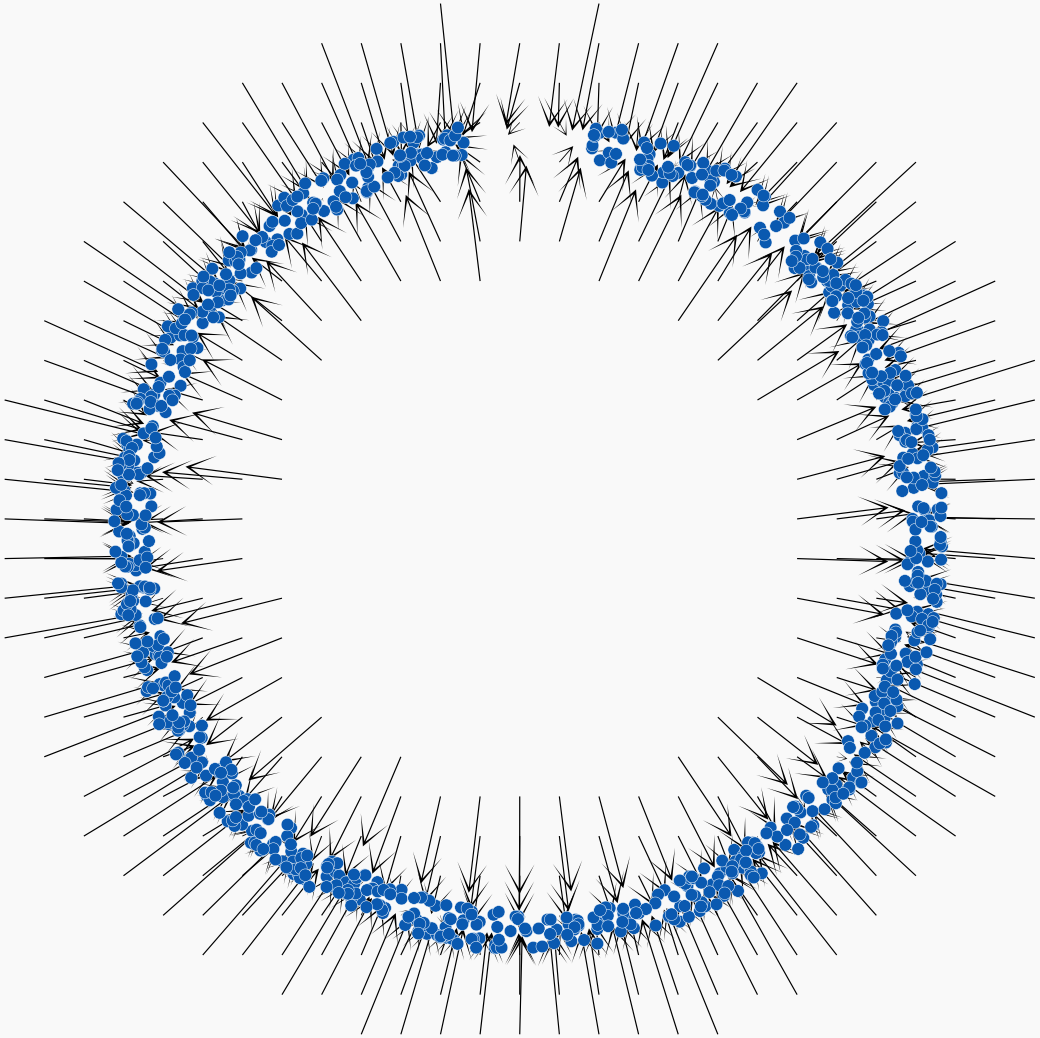
Reconstructed



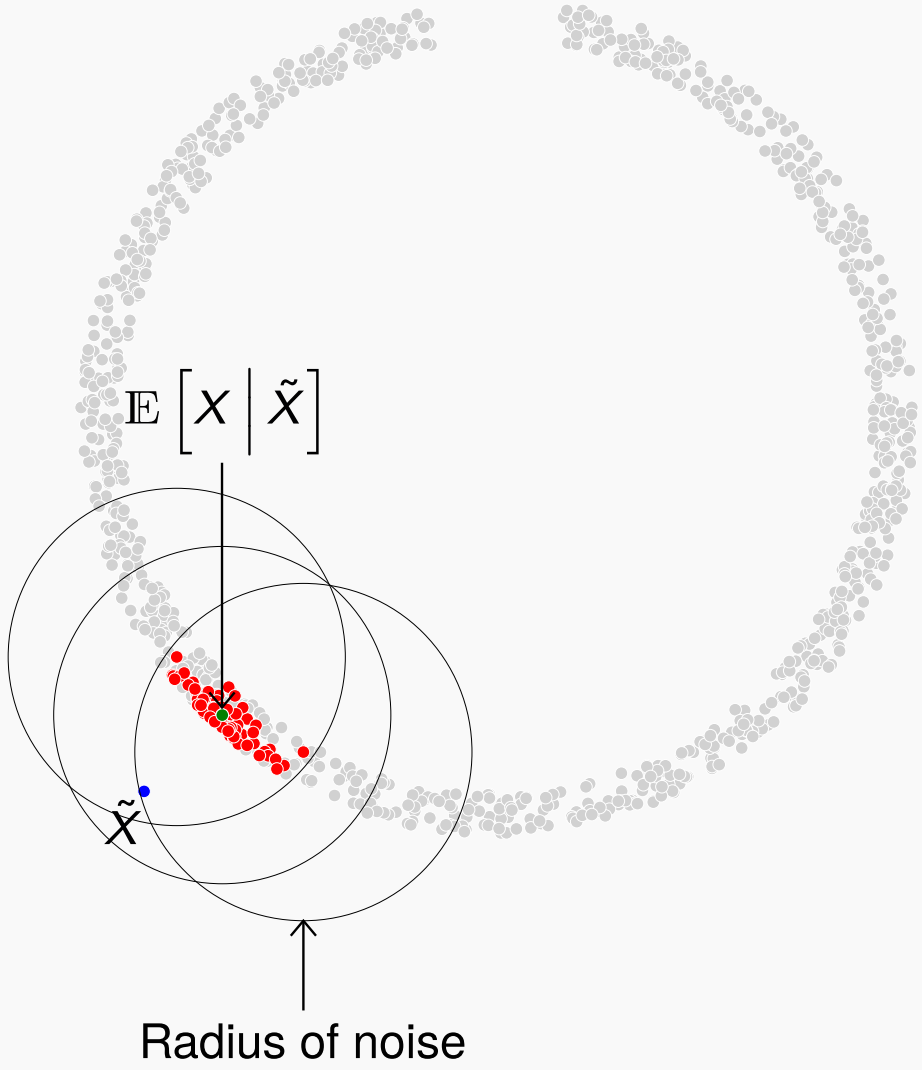
A key weakness of this approach is that the distribution of valid reconstructions given some corrupted inputs might be non-deterministic, possibly multi-modal.

The problem is that, when training with an MSE loss, the best reconstruction is the expected value of such a distribution of points.

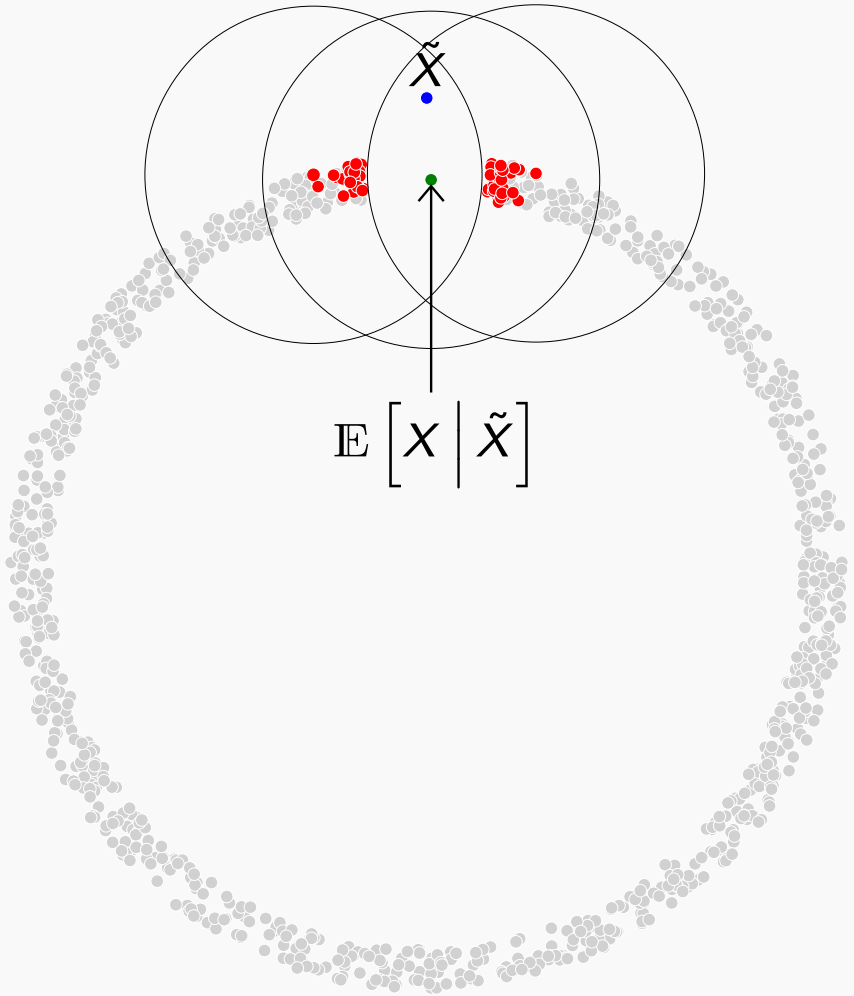




Problems with a multimodal $p(X|\tilde{X})$ can be visualized in the gap.

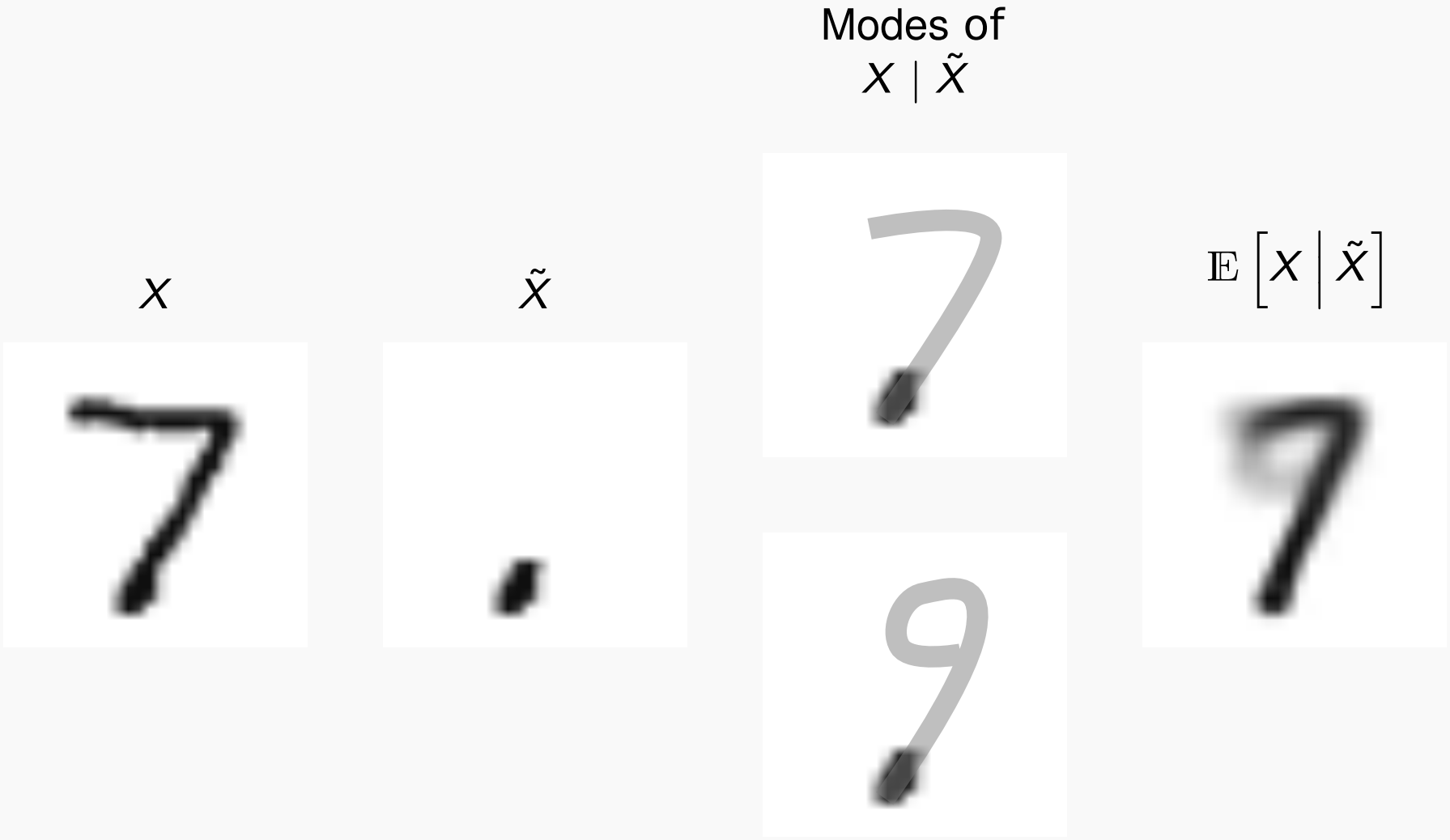


Problems with a multimodal $p(X|\tilde{X})$ can be visualized in the gap.



Problems with a multimodal $p(X|\tilde{X})$ can be visualized in the gap.

We can observe the same with very corrupted MNIST samples.



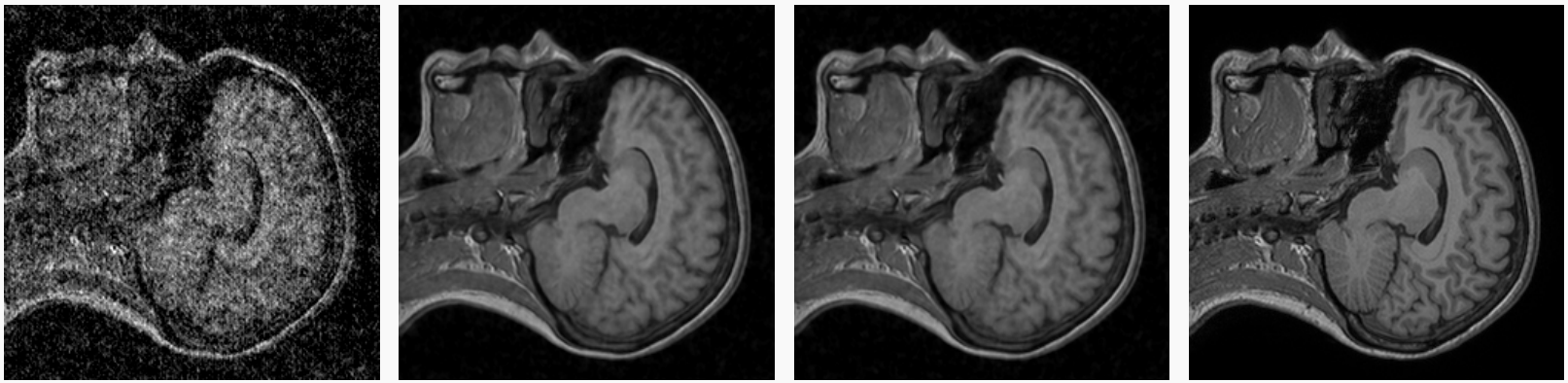
Denoising can be achieved even without clean samples, assuming the noise is additive and **unbiased**. Consider two **unbiased** independent sources of noise ϵ and δ. Then,

$$
\begin{aligned}
&\mathbb{E}\left[\|g(f(x+\epsilon)) - (x+\delta)\|^2\right] \\
&\quad = \mathbb{E}\left[\|(g(f(x+\epsilon) - x) - \delta)\|^2\right] \\
&\quad = \mathbb{E}\left[\|g(f(x+\epsilon)) - x\|^2\right] - 2\mathbb{E}\left[\delta^\top g(f(x+\epsilon)) - x\right] + \mathbb{E}\left[\|\delta\|^2\right] \\
&\quad = \mathbb{E}\left[\|g(f(x+\epsilon)) - x\|^2\right] - 2\underbrace{\mathbb{E}\left[\delta\right]^\top}_{=0}\mathbb{E}\left[g(f(x+\epsilon)) - x\right] + \mathbb{E}\left[\|\delta\|^2\right] \\
&\quad = \mathbb{E}\left[\|g(f(x+\epsilon)) - x\|^2\right] + \mathbb{E}\left[\|\delta\|^2\right].
\end{aligned}
$$

Hence,

$$\operatorname*{argmin}_{\theta} \mathbb{E}\left[\|g(f(x+\epsilon)) - (x+\delta)\|^2\right] = \operatorname*{argmin}_{\theta} \mathbb{E}\left[\|g(f(x+\epsilon)) - x\|^2\right]$$

solution is not dependent on whether ground-truth is noisy or not!



(a) Input 18.93 dB | (b) Noisy trg. 29.77 dB | (c) Clean trg. 29.81 dB | (d) Reference

Lehtinen et al., 2018

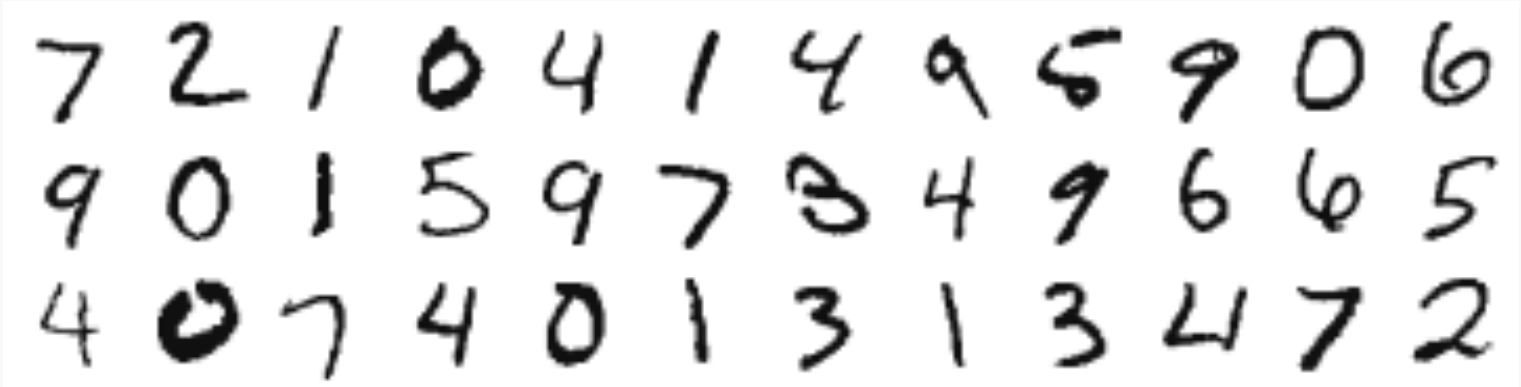
As a special case of denoising, we can corrupt inputs by downscaling the image and attempt to restore the original resolution. This is referred to as *super-resolution*.

We can do this by having an encoder whose input is smaller than the decoder's output.

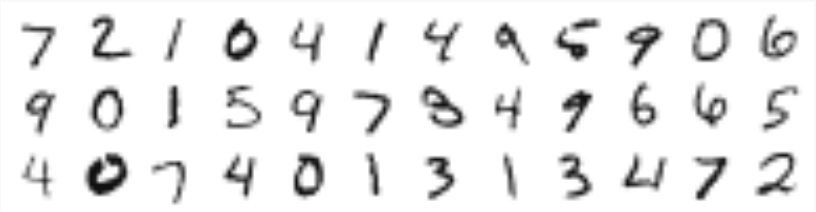
```
inputs = F.avg_pool2d(original, kernel_size=2)
z = encoder(inputs)
outputs = decoder(z)
# `outputs' has the same shape as `original'
loss = mse_loss(outputs, original)
```

```
MNISTUpscaler(
  (encoder): Sequential(
    (0): Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1))
    (3): ReLU(inplace=True)
    (4): Conv2d(32, 32, kernel_size=(4, 4), stride=(1, 1))
    (5): ReLU(inplace=True)
    (6): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1))
  )
  (decoder): Sequential(
    (0): ConvTranspose2d(32, 32, kernel_size=(4, 4), stride=(1, 1))
    (1): ReLU(inplace=True)
    (2): ConvTranspose2d(32, 32, kernel_size=(3, 3), stride=(2, 2))
    (3): ReLU(inplace=True)
    (4): ConvTranspose2d(32, 32, kernel_size=(4, 4), stride=(2, 2))
    (5): ReLU(inplace=True)
    (6): ConvTranspose2d(32, 32, kernel_size=(5, 5), stride=(1, 1))
    (7): ReLU(inplace=True)
    (8): ConvTranspose2d(32, 1, kernel_size=(5, 5), stride=(1, 1))
  )
)
```

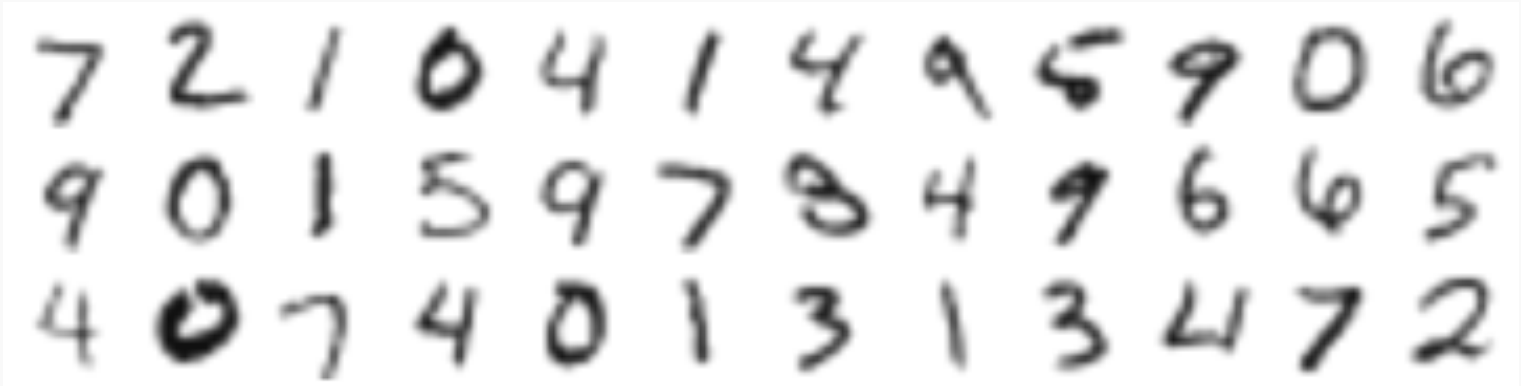
Original



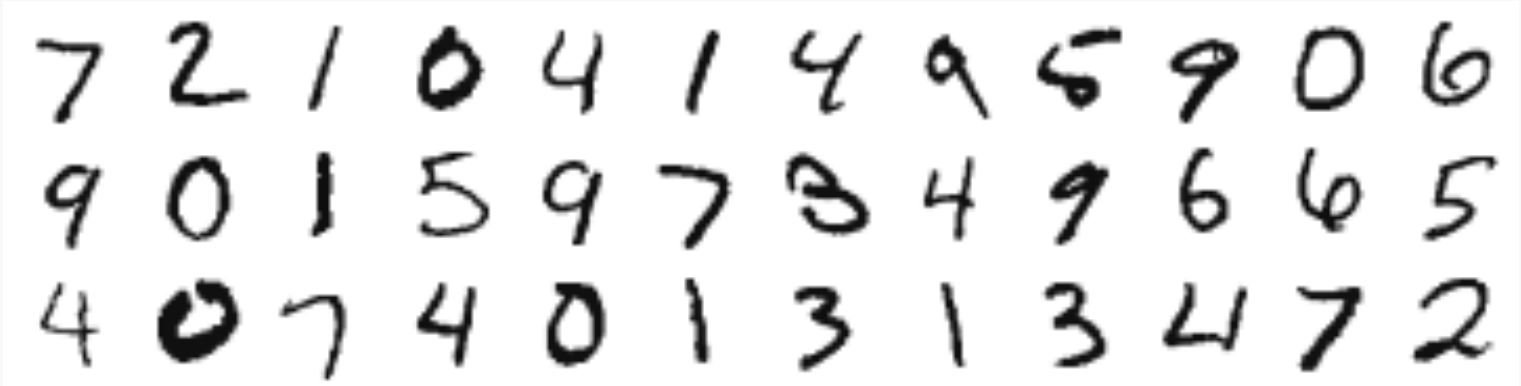
Input



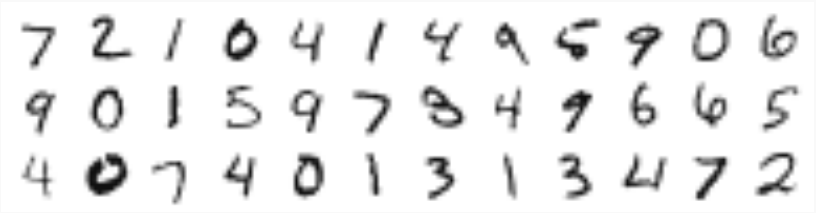
Bilinear interpolation



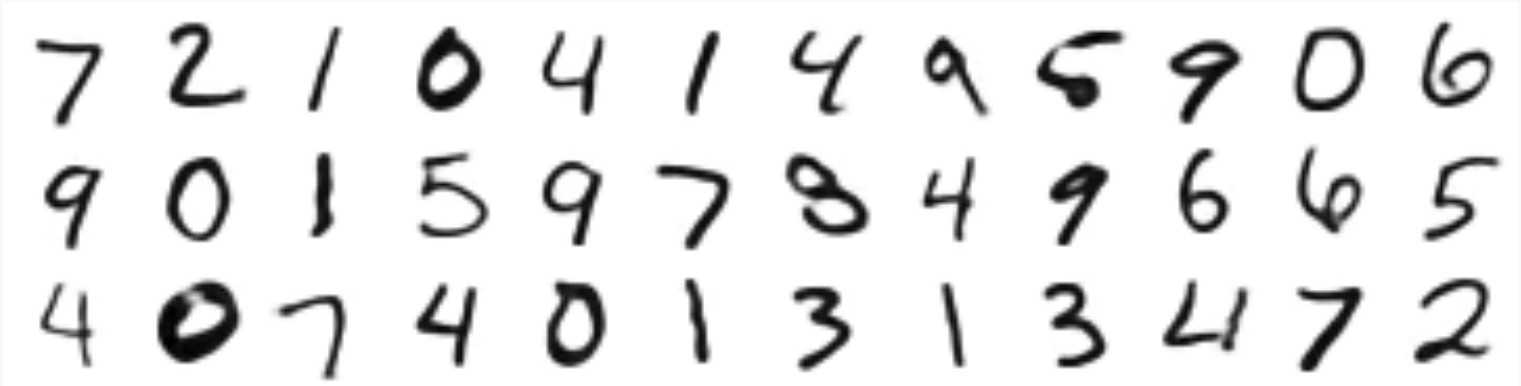
Original



Input



Autoencoder output



stop!

One theoretical grounding for autoencoders is seen from the maximization of the *mutual information* between inputs x and latent codes z (Vincent et al., 2010).

Let x be a sample, $z = f(x; \theta)$ its latent code, and $q(x, z)$ the joint distribution of x and z.

We can write

$$\begin{aligned}
\underset{\theta}{\operatorname{argmax}}\ \mathbb{I}(x; z) &= \underset{\theta}{\operatorname{argmax}}\ \underbrace{\mathbb{H}(x)}_{\text{Ignored by } \theta} - \mathbb{H}(x|z) \\
&= \underset{\theta}{\operatorname{argmax}}\ -\mathbb{H}(x|z) \\
&= \underset{\theta}{\operatorname{argmax}}\ \mathbb{E}_{q(x,z)} \left[\log q(x|z)\right].
\end{aligned}$$

For any distribution p, we have

$$\mathbb{E}_{q(x,z)}\left[\log q(x|z)\right] \geq \mathbb{E}_{q(x,z)}\left[\log p(x|z)\right],$$

since $D_{\mathrm{KL}}\left(q\|p\right) \geq 0$ (Task 23 (i)).

If we find a p which is "good enough" for our problem, then we have a good lower bound on the mutual information between x and z, which means the latent codes contain a lot of information from the inputs.

Consider the following model for p

$$p(\cdot|Z = z) = \mathcal{N}(g(z;\psi), \sigma),$$

where g is parametrized by ψ and σ is fixed. This will result in

$$\mathbb{E}_{q(x,z)}\left[\log p(x|z)\right] = -\frac{1}{2\sigma^2}\mathbb{E}_{q(x|z)}\left[\|x - g(f(x;\theta);\psi)\|^2\right] + k,$$

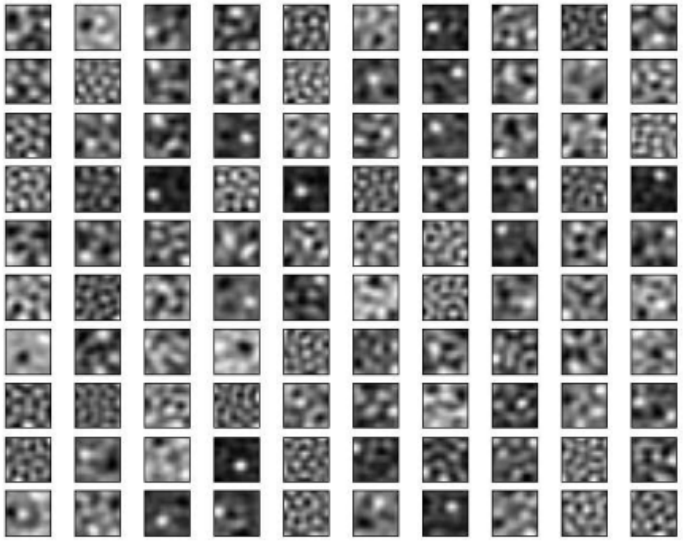
for some constant k.

If we optimize ψ for making the reconstructions close to x, the loss becomes:

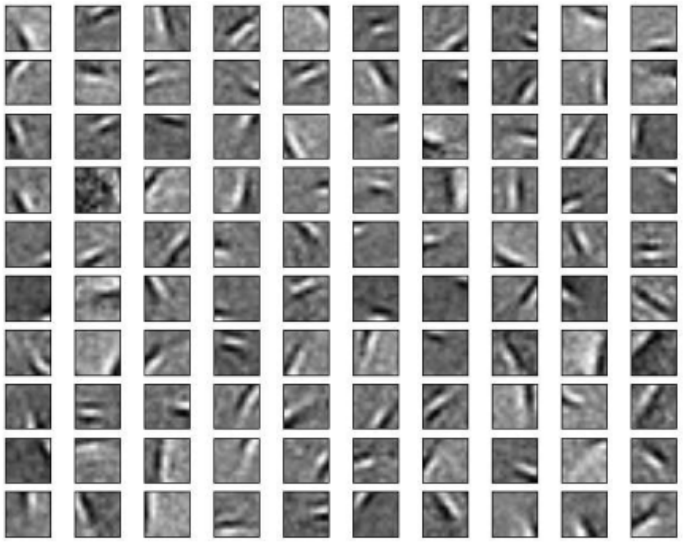
$$\operatorname*{argmax}_{\theta} \mathbb{I}(x;z) \approx \operatorname*{argmin}_{\theta}\left(\min_{\psi} \frac{1}{N}\sum_{n=1}^{N} \|x_n - g(f(x_n;\theta);\psi)\|^2\right).$$

While this provides an interpretation of where the autoencoder objective comes from and what it implies, maximization of mutual information is not enough for obtaining good latent codes. In fact, one could simply have an identity encoder, which maximizes mutual information, but is clearly not useful.

This observation justifies constraining the latent codes to a dimensionality lower than the input dimensionality. Furthermore, good representations should be robust under noisy inputs, meaning a denoising objective encourages the model to separate the useful signal from noise.



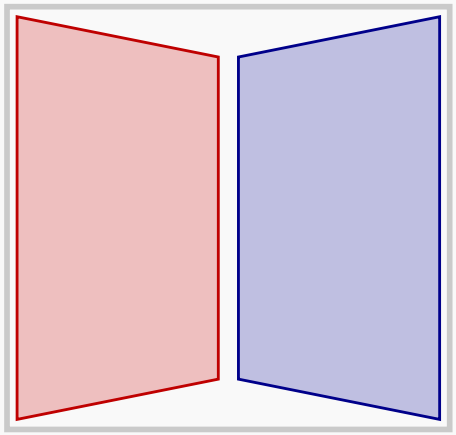
L2-regularized



Denoising

Since they learn to extract useful representations, it is naturally possible to use learned encoders as feature extractors for other machine learning tasks.

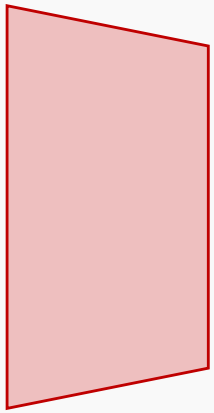
For instance, one could use them to initialize the layers of deep neural network classifiers.



Autoencoder

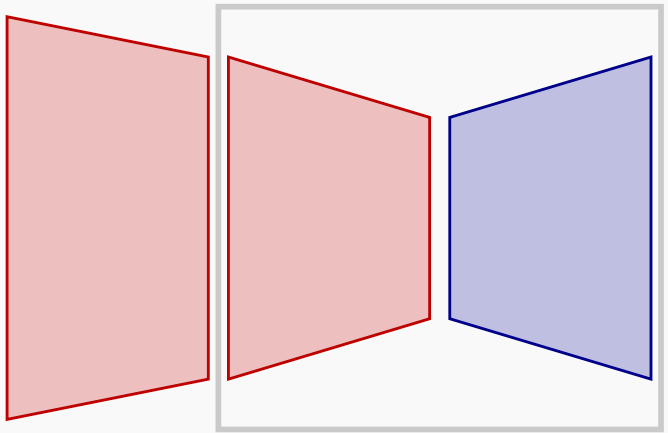
Since they learn to extract useful representations, it is naturally possible to use learned encoders as feature extractors for other machine learning tasks.

For instance, one could use them to initialize the layers of deep neural network classifiers.



Since they learn to extract useful representations, it is naturally possible to use learned encoders as feature extractors for other machine learning tasks.

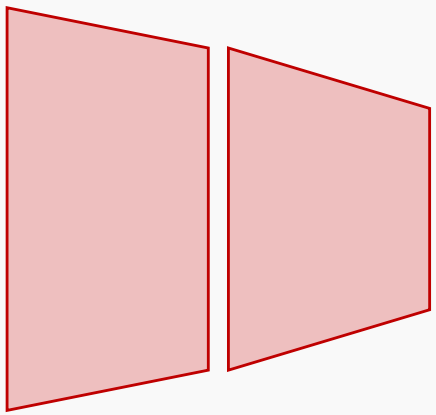
For instance, one could use them to initialize the layers of deep neural network classifiers.



Autoencoder

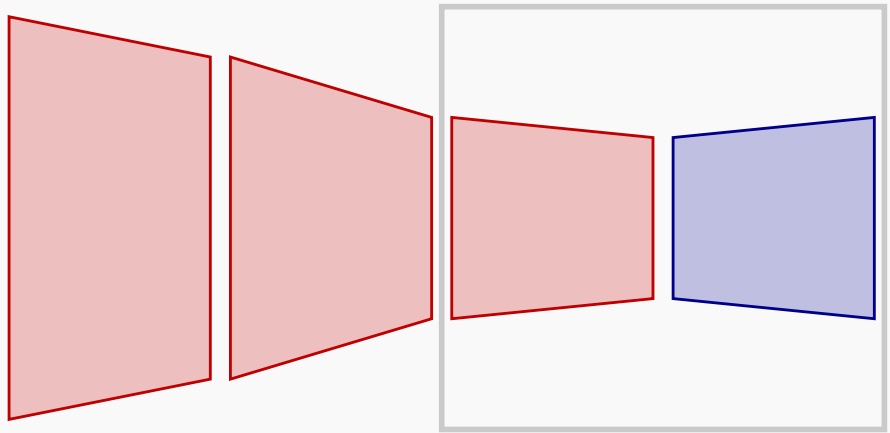
Since they learn to extract useful representations, it is naturally possible to use learned encoders as feature extractors for other machine learning tasks.

For instance, one could use them to initialize the layers of deep neural network classifiers.



Since they learn to extract useful representations, it is naturally possible to use learned encoders as feature extractors for other machine learning tasks.

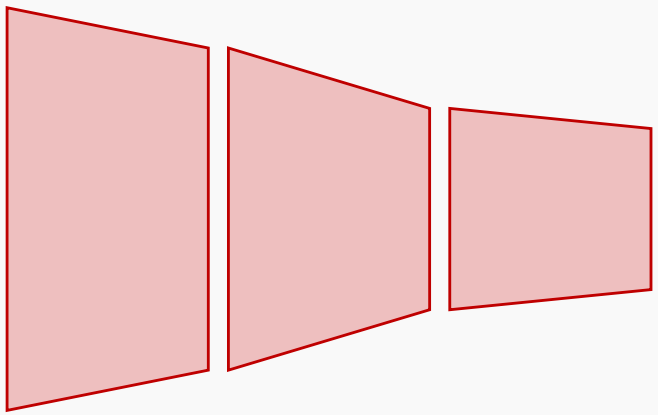
For instance, one could use them to initialize the layers of deep neural network classifiers.



Autoencoder

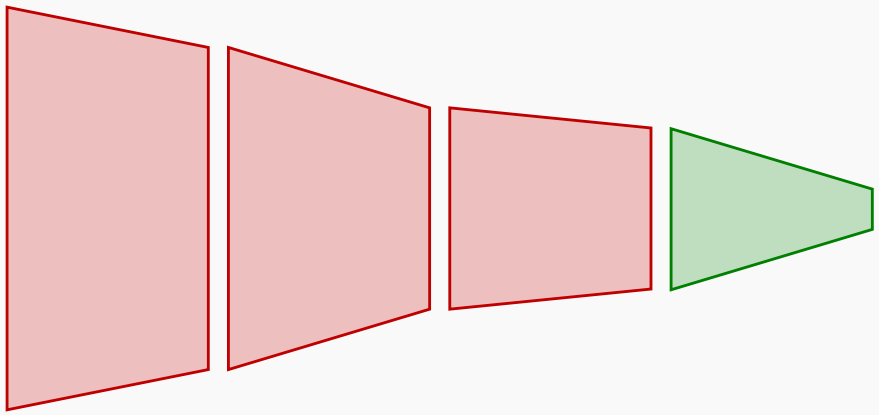
Since they learn to extract useful representations, it is naturally possible to use learned encoders as feature extractors for other machine learning tasks.

For instance, one could use them to initialize the layers of deep neural network classifiers.



Since they learn to extract useful representations, it is naturally possible to use learned encoders as feature extractors for other machine learning tasks.

For instance, one could use them to initialize the layers of deep neural network classifiers.



a final classification layer is added and the whole structure can be fine-tuned.

References

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., and Aila, T. (2018). Noise2noise: Learning image restoration without clean data. arXiv preprint arXiv:1803.04189.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., and Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12).