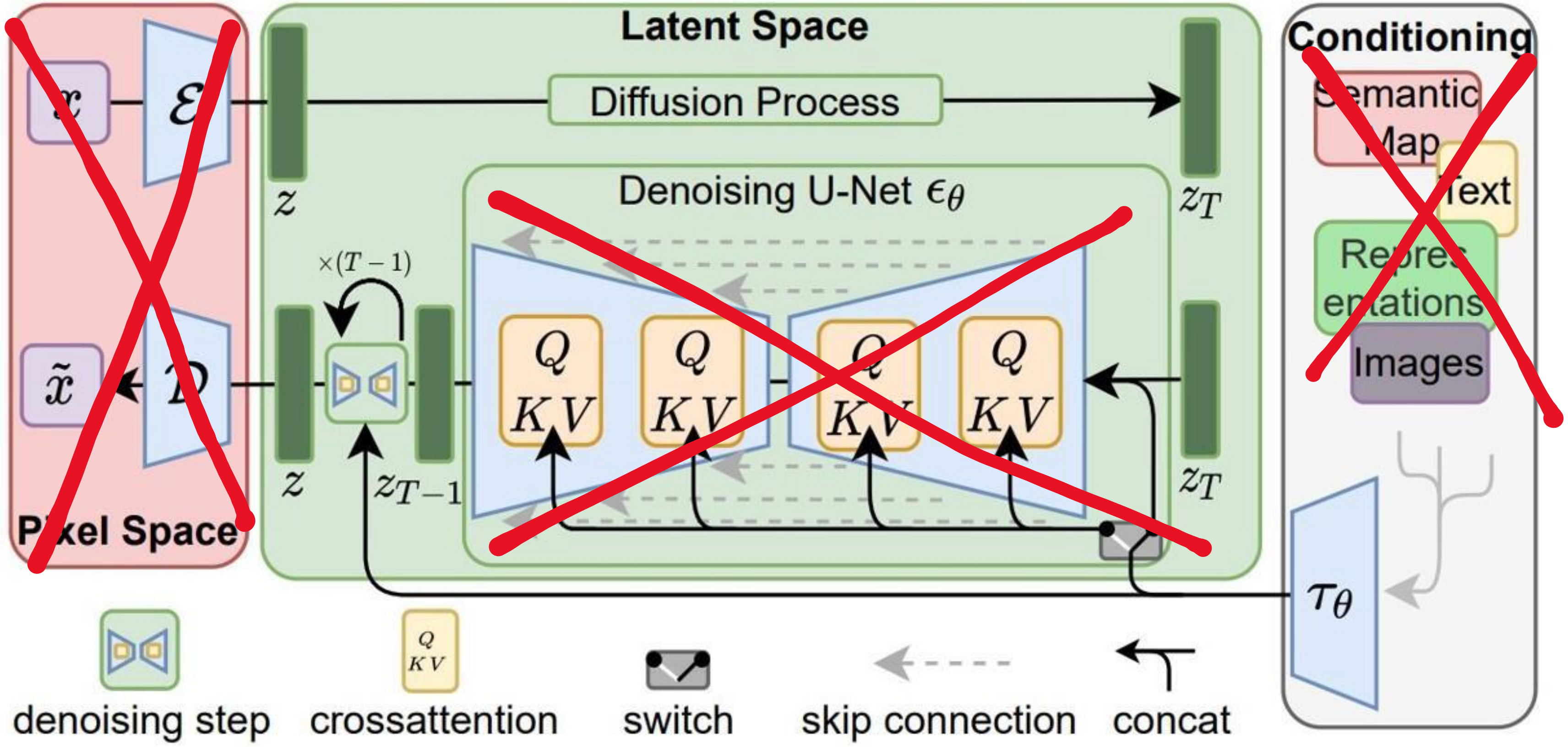
DDPM

Diffusion

Back to this…



ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

Diffusion

Here is the algorithm for di

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
  $$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
until converged

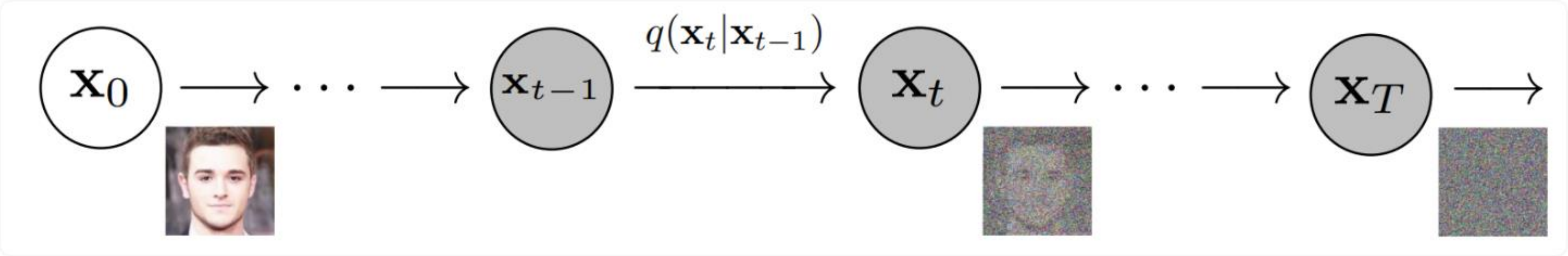
Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
end for
return $\mathbf{x}_0$

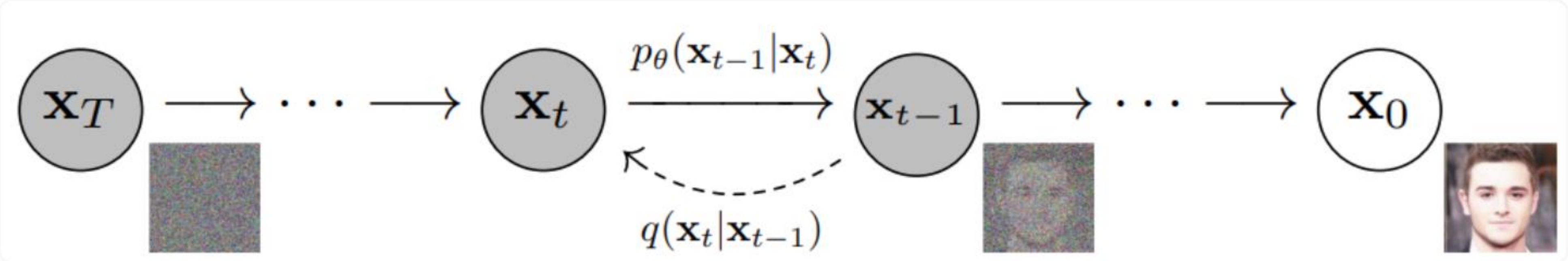
Diffusion

Diffusion models have a forward process and a reverse process...

- Forward:



- Reverse:



ACCELERATE PROGRAMME FOR SCIENTIFIC DISCOVERY

SCHMIDT FUTURES

Diffusion

The choice of distribution makes a few things easy for us...

- The amount of noise added at each step is defined by a ***variance schedule*** (usually called β):

- A nice property of the normal distribution is that we can sample at an arbitrary time step in closed form using a neat trick.
- Essentially,

$$\bar{\boldsymbol{x}}_t = \sqrt{\bar{\alpha}_t}\,\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$$

- During the reverse sampling process, we then make use of similar tricks and about a page of maths, to say that *the mean of noise component at any given time step is*

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right)$$

Diffusion

What does this mean?

- We now know that we can simply take a random image, and add an amount of noise according to our variance schedule.
- Instead of having our model predict exactly what the image is (i.e. denoise in one step), we have the model predict the amount of noise added at a particular timestep.
- We then subtract this noise from the image, resulting in a slightly less noisy image.
- During training, we are ensuring that the model pays very careful attention to when only a very little amount of noise is added, and takes small steps, whereas when much noise is added, we want to model to take bigger steps.
- During sample, when we subtract noise from the image, we reinsert the less noisy image back into the model and repeat the process for a certain number of steps.

Diffusion

Here is the algorithm for diffusion…again…

Algorithm 1 Training

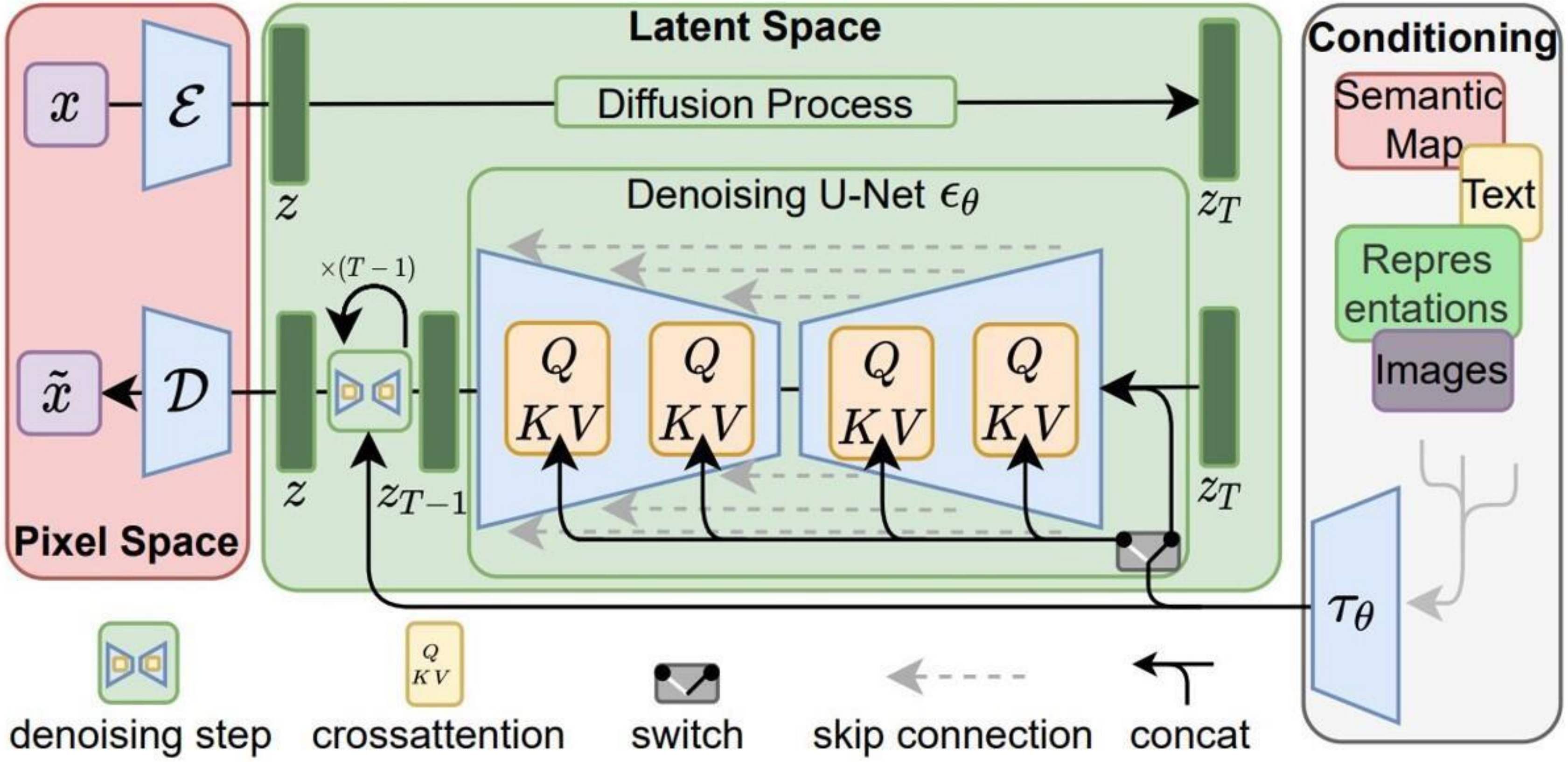
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Diffusion

Perhaps this makes more sense now…



Diffusion

Diffusion model paradigms – (un)conditional and classifier(-free)

- When we talk about “conditioning” a model, we are usually referring to additional information fed to the model
- Unconditional generation is simply following the DDPM algorithm with no text labels.
- But we can also just generate content without any guidance – this is unconditional generation
- There are a few main types of conditioning mechanisms:
  - Concatenation
  - Gradient-based
  - Cross-attention
- Classifier vs classifier-free
  - *Classifier* – train an additional classifier to guide the model.
    - Complicated, begin to approach GANs.
  - Classifier-free – mix the score estimates of a conditional model and an unconditional model.

Diffusion

Classifier-free Guidance

- Classifier-free Diffusion Guidance, Ho and Salimans, Google, 2022 ~1800 citations
- Encode the labels and pass them into the model but randomly drop some elements…

What…? Why…?

- Because now, we predict two types of noise – the conditional and the unconditional noise
- Subtract them from one another so that we can see the noise associated with the context:

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda)$$

- We can then use w to exaggerate the context. When $w = 0$, the model is entirely conditional. When w is strongly negative, this is saying: *DO NOT produce this class.*
- This allows for a balance between creativity and specificity.