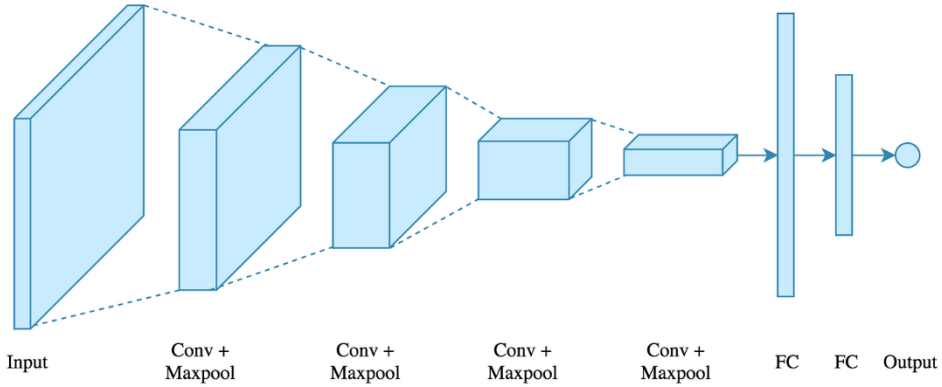
Transformers and foundation models

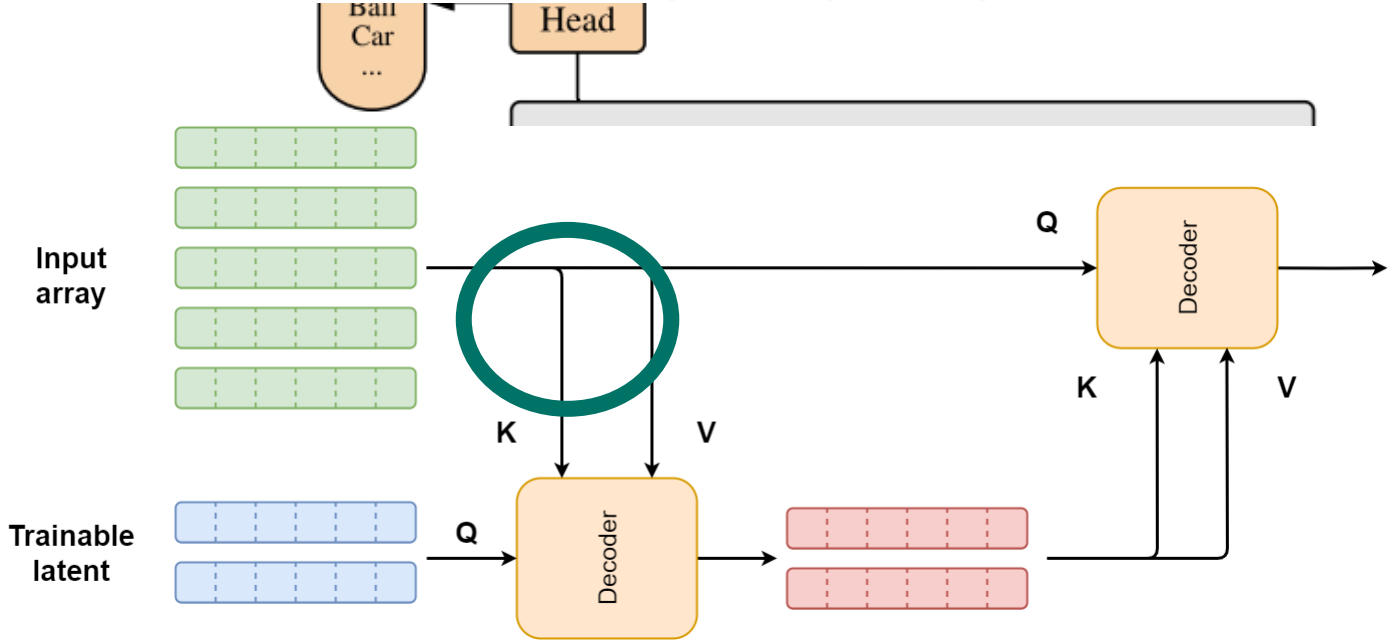
Malte Algren

Transformers

Set-to-set mapping

- A transformer takes in a set and returns a set
- Classification/regression: vector or scalar
- Add trainable latent space
  - Return that point at the end
  - Class token



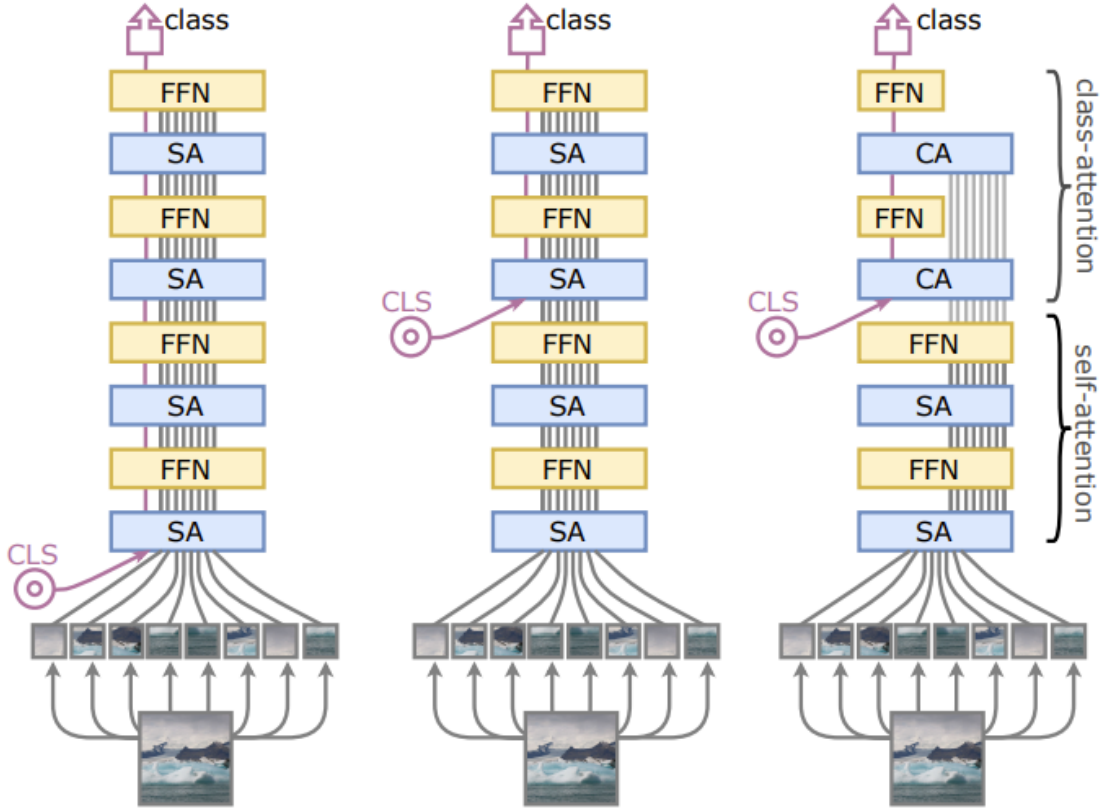


Transformers

Set-to-set mapping

- A transformer takes in a set and returns a set.
- Classification/regression: vector or scalar
- Add trainable latent space
  - Return that point at the end
  - Class token

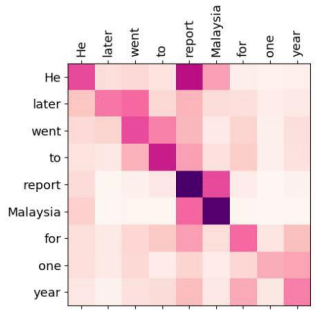
Going deeper with transformers

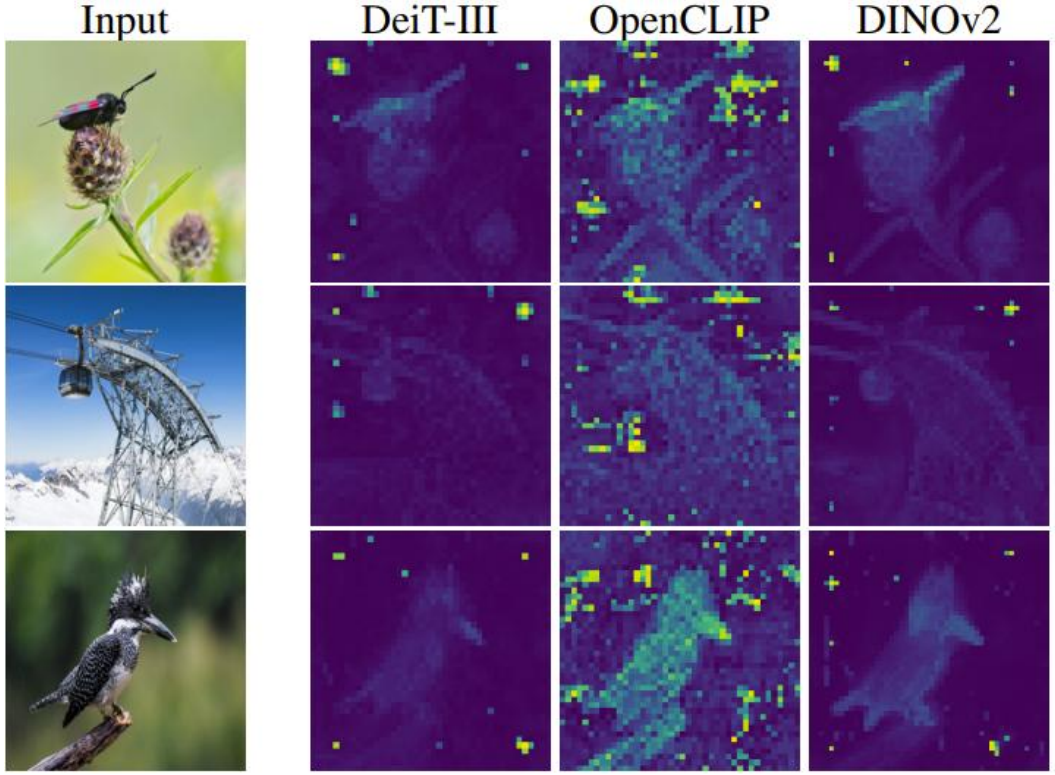


Transformers

Set-to-set mapping

- Full self attention also create a lot of junk
  - Where to send the junk?

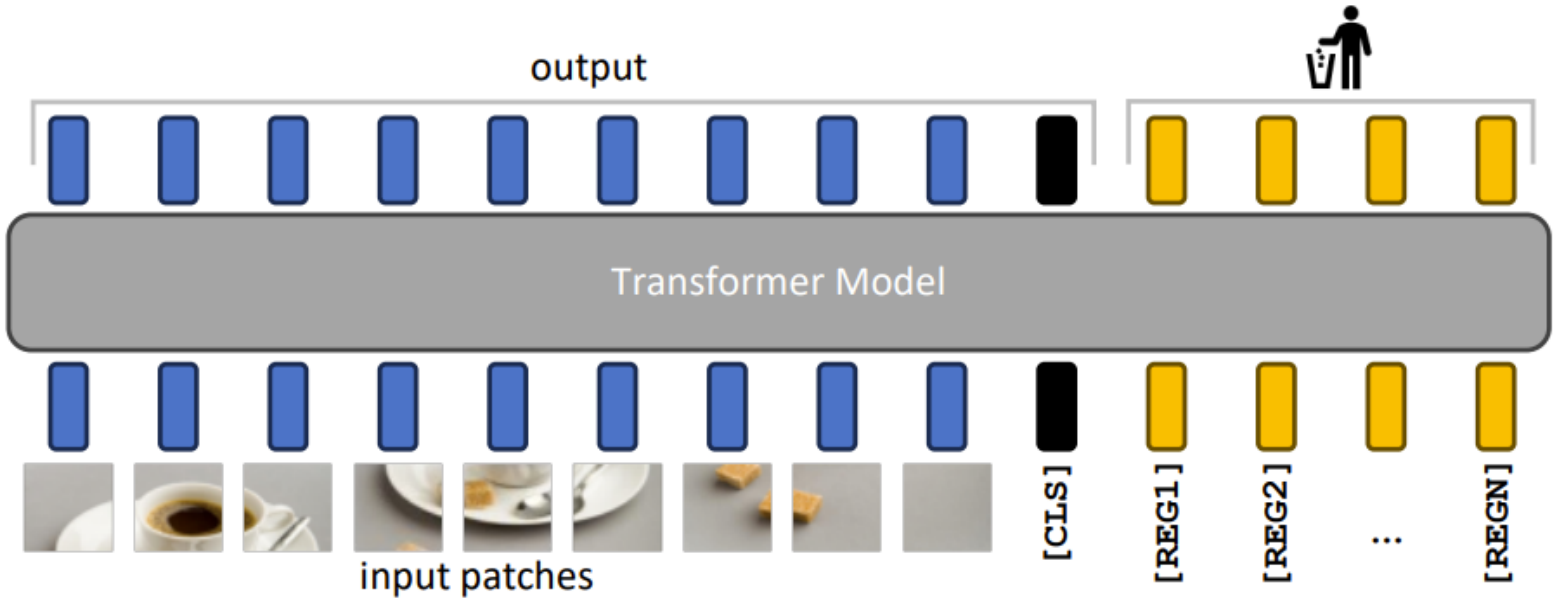




Transformers

Set-to-set mapping

- Full self attention also create a lot of junk
  - Where to send the junk?
  - Garbage collectors

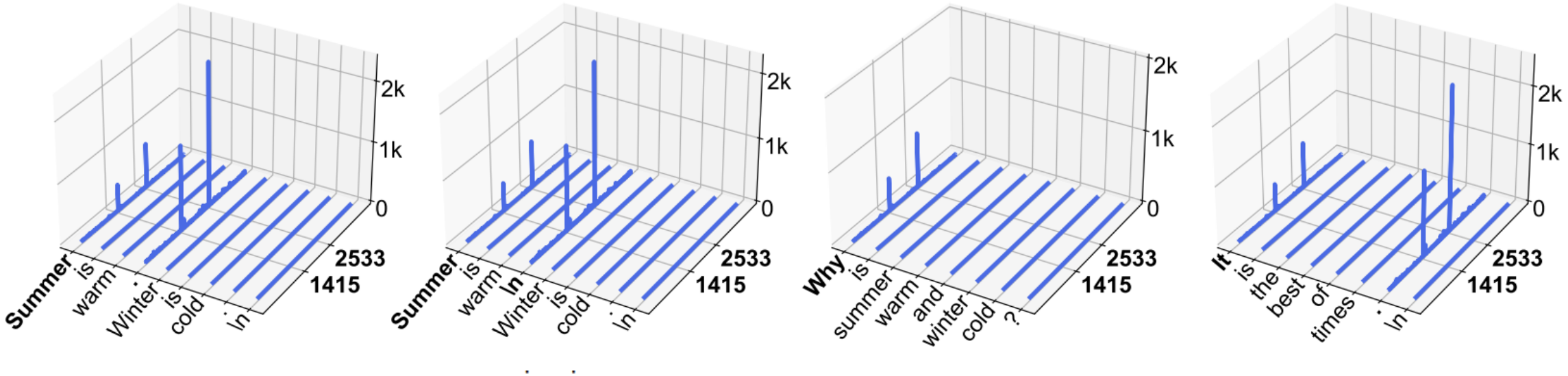


Transformers

Set-to-set mapping

- Full self attention also create a lot of junk
  - Where to send the junk?
- Adding trainable token that will be discarded afterwards

Questions?



Foundation models
How to make pretraining sound sexy

Malte Algren

Zoo of foundation models

Google trends on “Foundation model” since 2020:

- Word2vec & GloVe (2013)
- GPT (2018)
- BERT/BEiT (2018)
- DALL-E (2021)

[A Survey on Self-supervised Learning](#) (2023)

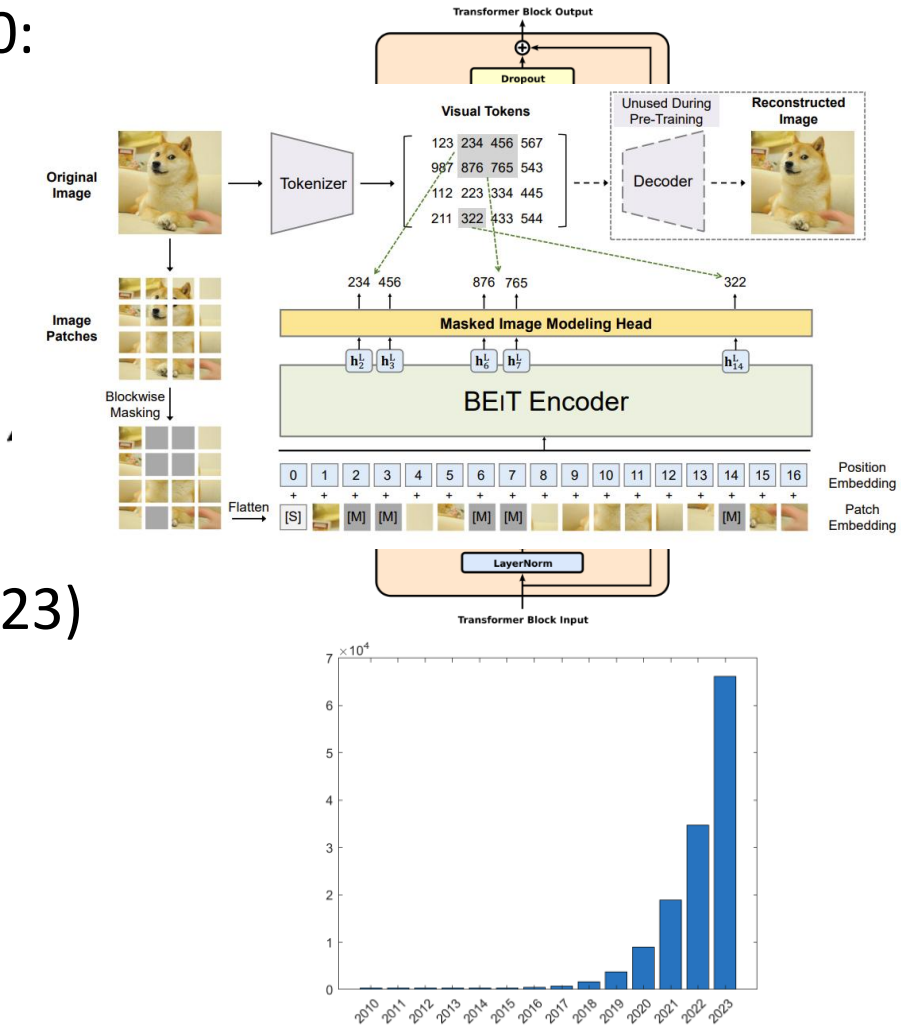


Fig. 2: Google Scholar search results for “self-supervised learning”. The vertical and horizontal axes denote the number of SSL publications and the year, respectively.

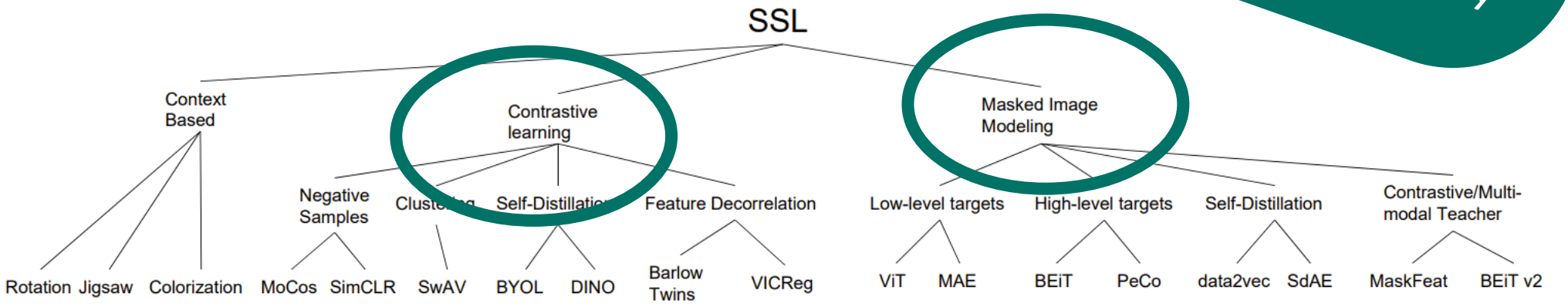
Zoo of foundation models

Google trends on “Foundation model” since 2020:

- Word2vec & GloVe (2013)
- GPT (2018)
- BERT/BEiT (2018)
- DALL-E (2021)

A Survey on Self-supervised Learning (2023)

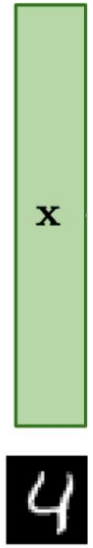
These concepts connects nicely with what we talked about Wednesday



Notation and definitions

Need to define some notation and language for foundation models

1. Backbone (encoder)
   - $g(x) = z \Longrightarrow f(z) = y'$
2. Latent representation
   - Representation: z
3. Learning objective
   - Supervised: $\mathcal{L}(f(z), y)$
   - Self-supervised: $\mathcal{L}(f(z), x)$



(V)AE (1990s)

Notation and definitions

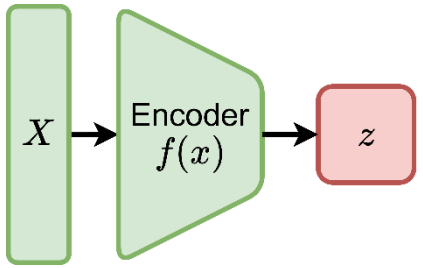
Need to define some notation and language for foundation models

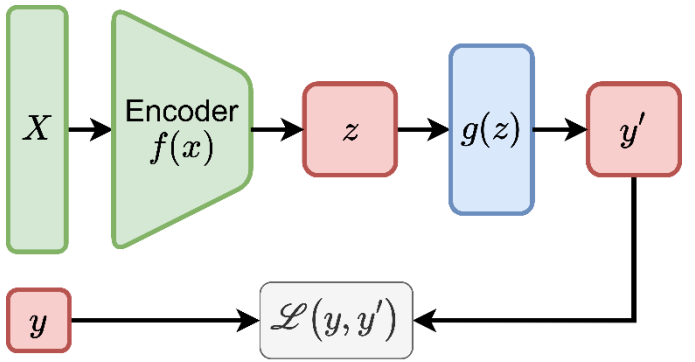
1. Backbone (encoder)
   - $g(x) = z \Longrightarrow f(z) = y'$
2. Latent representation
   - Representation: z
3. Learning objective
   - Supervised: $\mathcal{L}(f(z), y)$
   - Self-supervised: $\mathcal{L}(f(z), x)$
4. Scale (The world biggest heist)
5. Downstream tasks $T_n(z)$

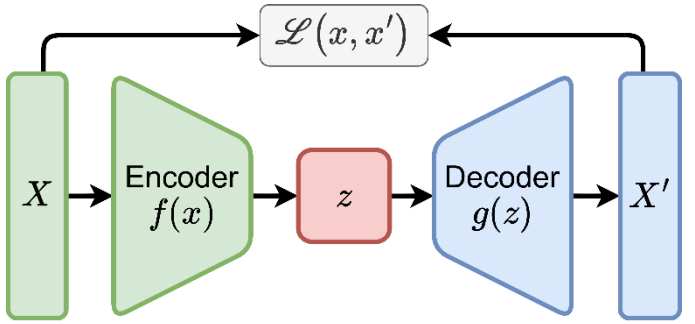
Subset of the lawsuits

Pretraining method

- What does pretrain mean?
- Constructing a pretrain model to solve tasks $T_n(z)$
  - What is the object of a pretrained model?
    1. $T(z) \approx p(y)$
    2. $T(z) \approx p(x)$
    3. $T(z) \neq p(x) \vee p(y)$
    4. $T(z) \neq\neq\neq\neq\neq p(x) \vee p(y)$
  - Often $T(z)$ is unknown but assume $T(z) \approx p(x)$

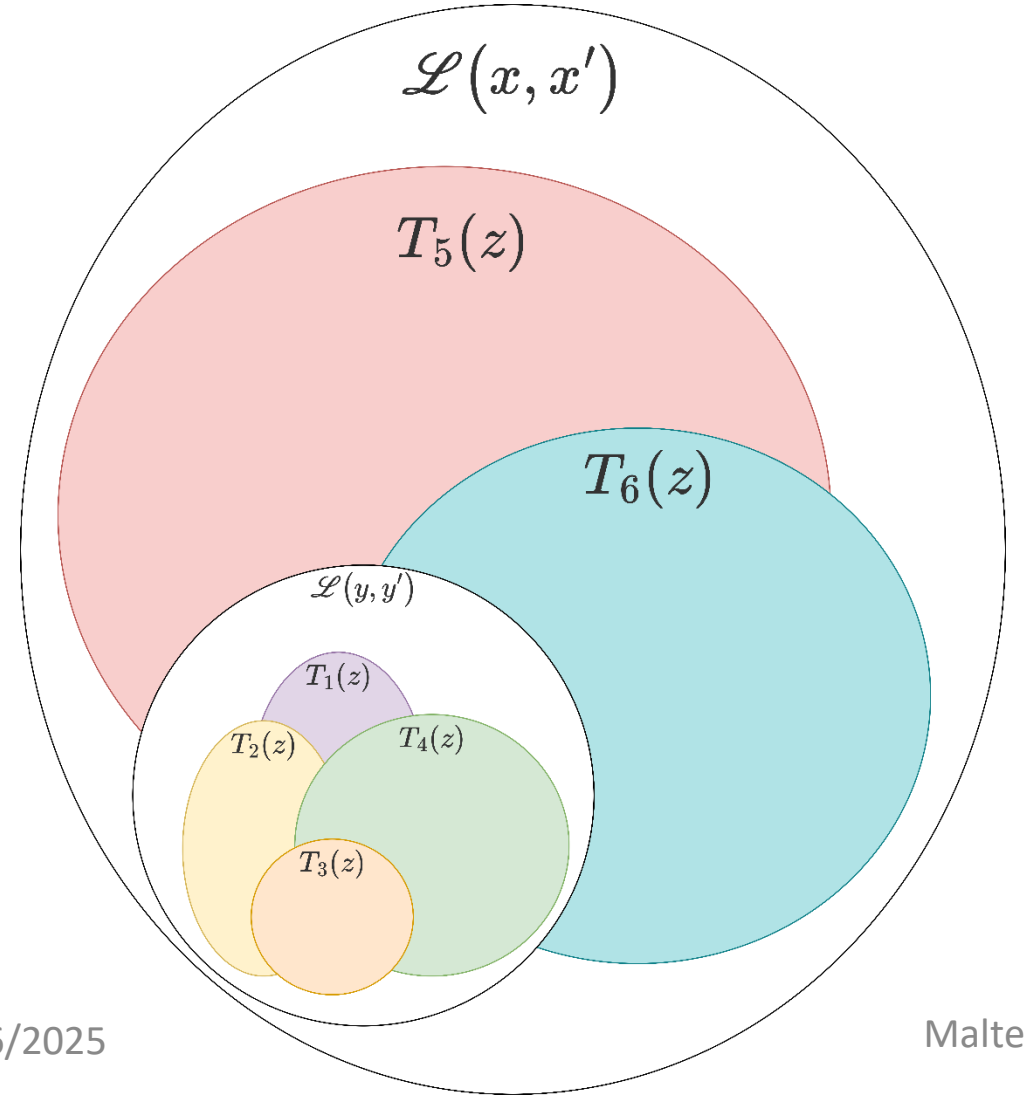


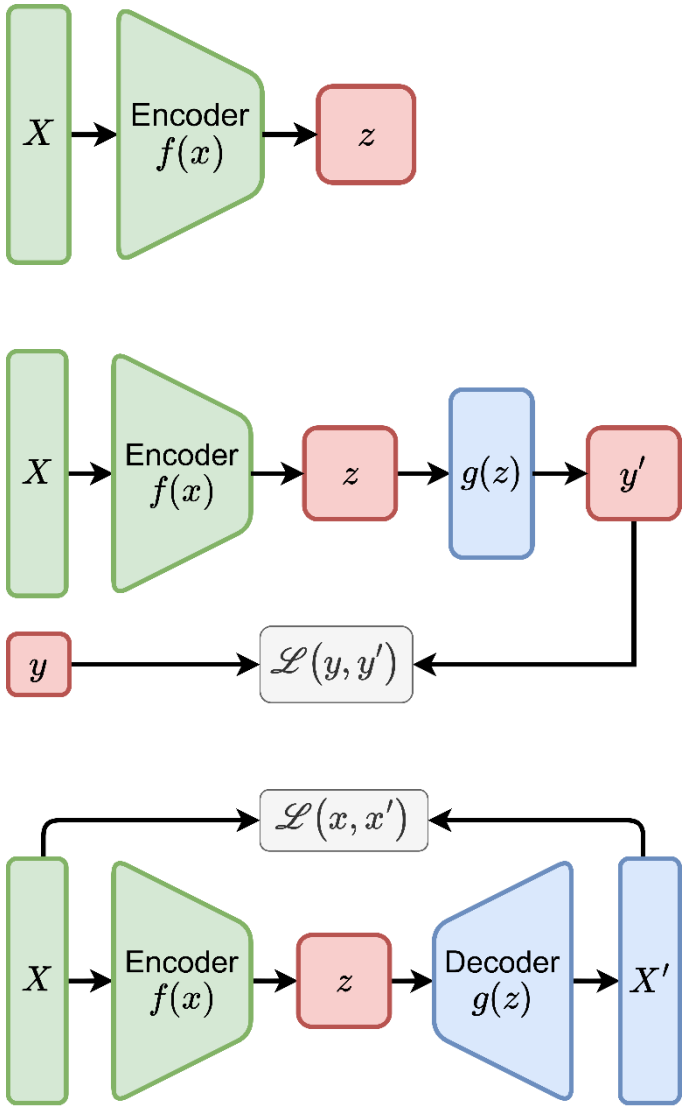




Pretraining method

- Constructing a pretrain model to solve tasks $T_n(z)$

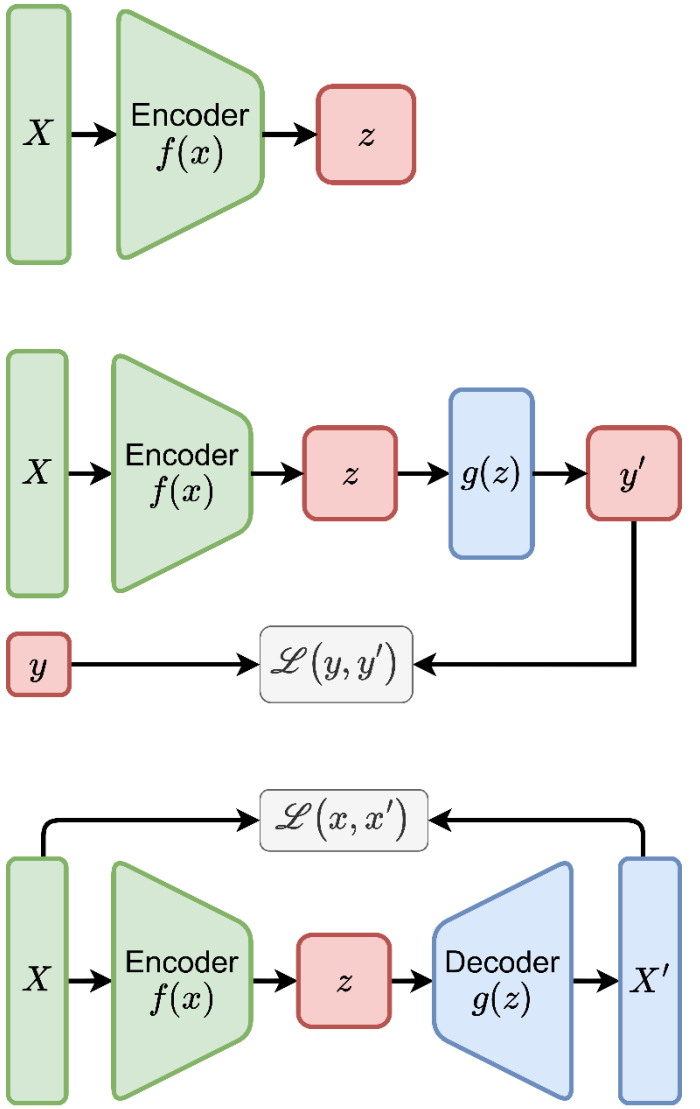




Pretraining method

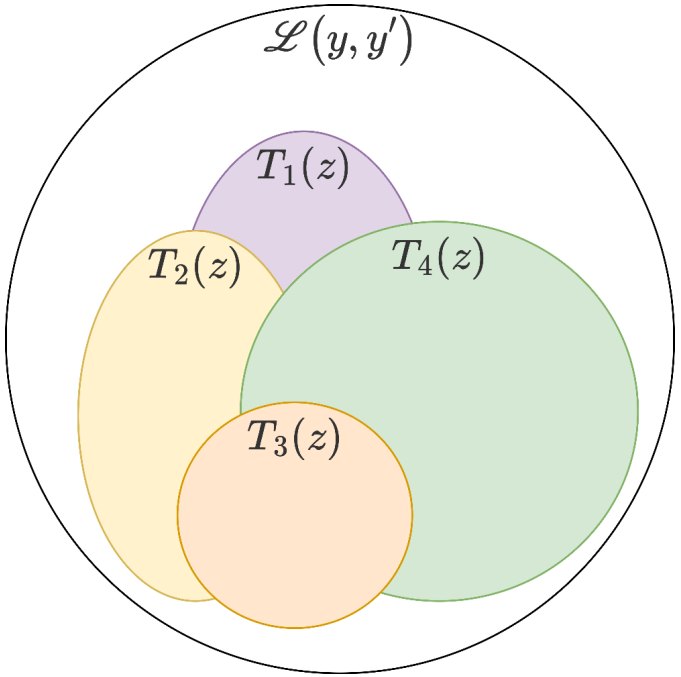
- Constructing a pretrain model to solve tasks $T_n(z)$
  1. Supervised learning $T(z) \approx p(y)$
     - MSE, BCE, CE, etc.
     - Multi-task learning with σ
  2. Self-supervised $T(z) \approx p(x)$
  3. Semi-supervised $T(z) \approx p(x, y)$
     - Joint supervised and self-supervised

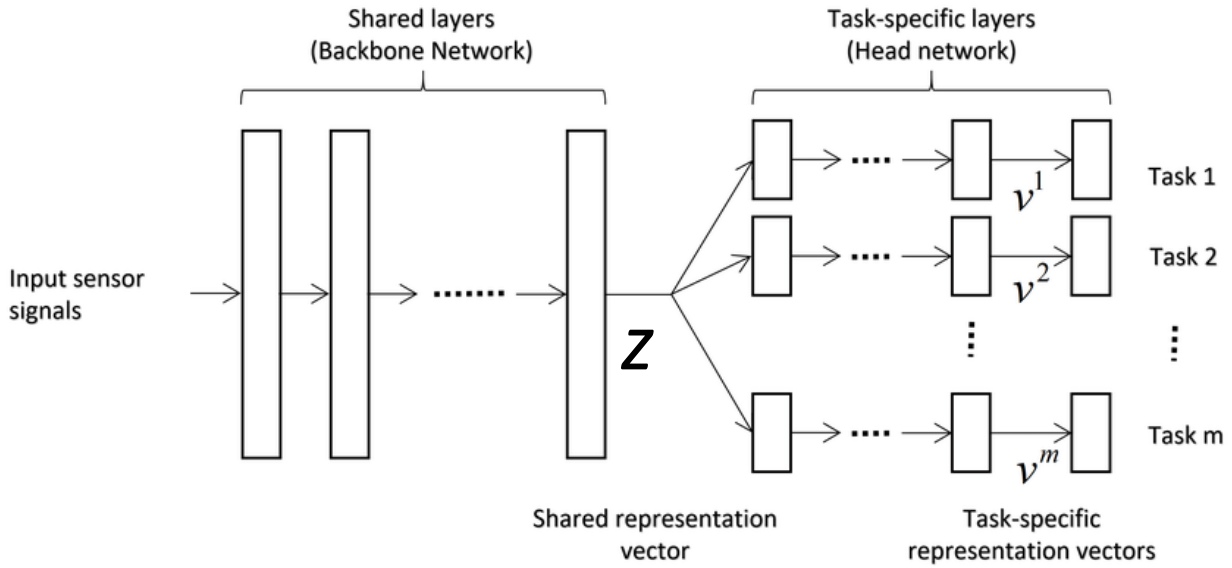
  A. Final step: Finetuning



Supervised learning

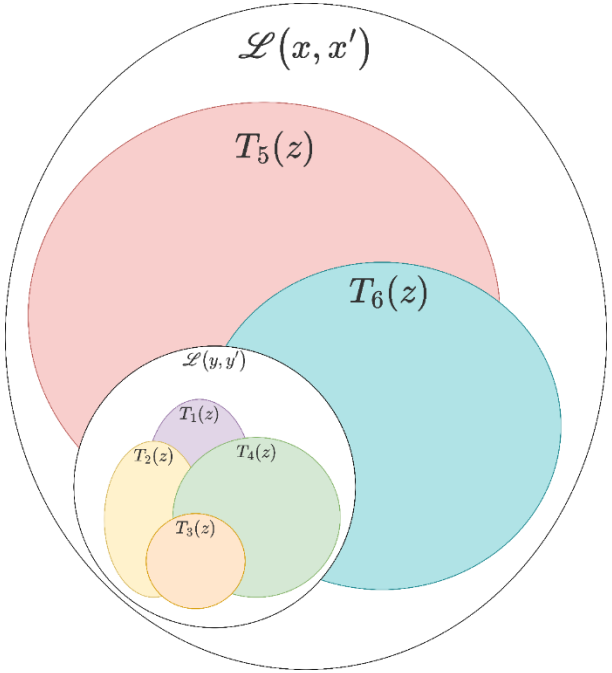
- If you have an idea about our downstream tasks
  - Use *x* to model *y* – access to *y*
  - Multiple downstream task that are orthogonal
  - Board support in the phase space
- Construct your representation *z* using $T_n(z)$

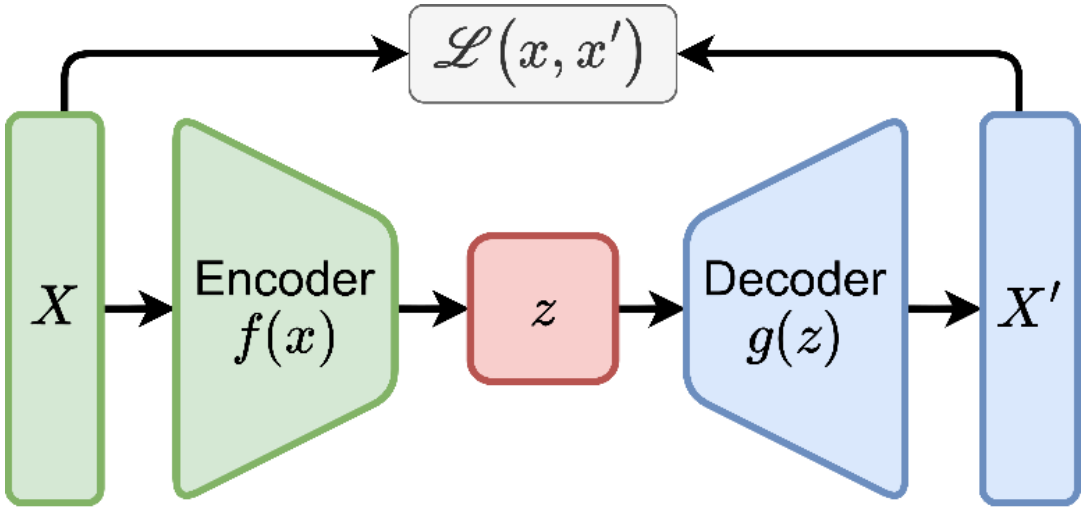




Self-supervised learning

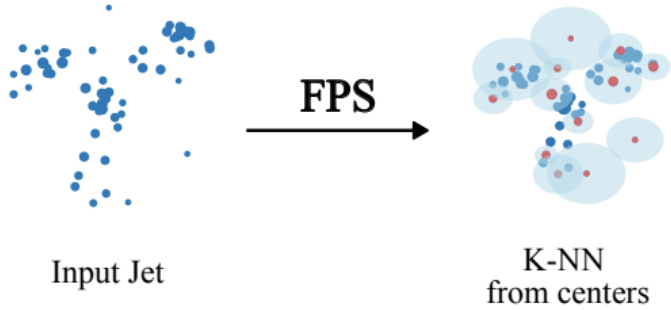
- You don't have any *y*. You have access to *x*
  - No specific $T_n(z)$ in mind
  - Board support in the phase space
  - Always similar encoder/decoder setup
  - In physics we have a lot of record data but no labels





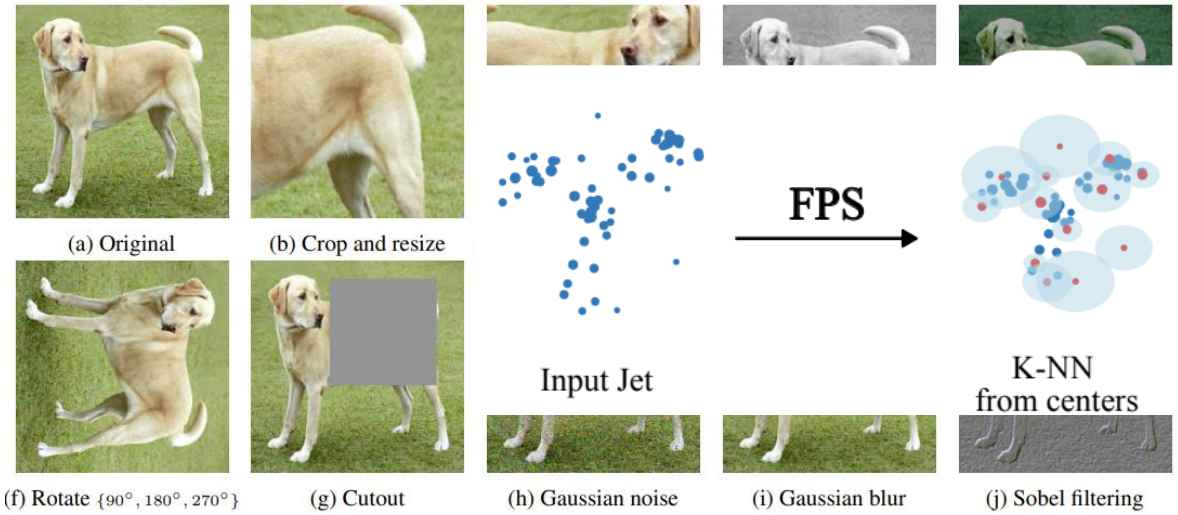
Self-supervised learning

- Constructing a task from samples *X*
- Augmentations is the name of the game
  - $A(x) = x_a \Rightarrow \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(f(x_a), x)$
  1. Masking
  2. Cropping/cutout
  3. Rotation
  4. Other invariances
- Many of them are domain specific



Self-supervised learning

- Constructing a task from samples *X*
- Augmentations is the name of the game
  - Masking (general)
  - Cropping (image)



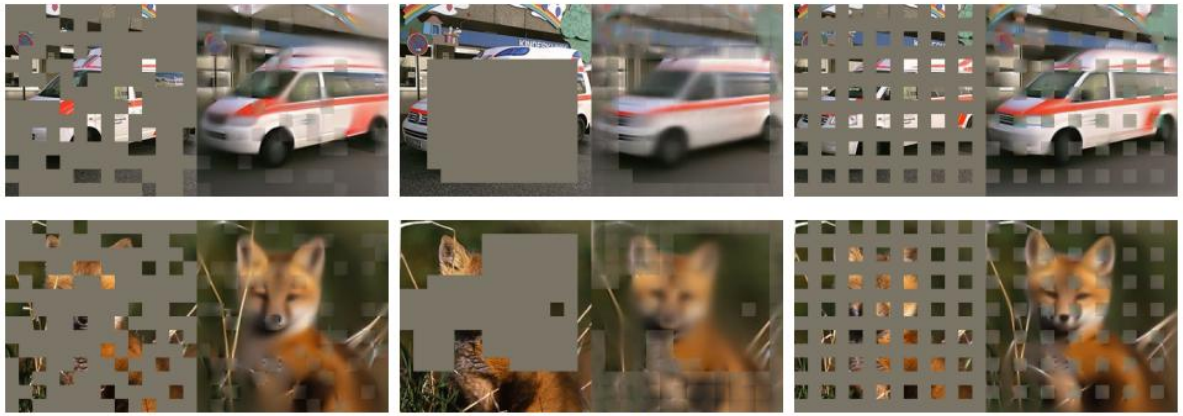
Generative	Joint embedding - CLR	Compression or not?
1. Masked autoencoder MAE	1. SimCLR	What do we want?
2. Diffusion MAE	2. BYOL	Representation learning vs pretraining
	3. I-JEPA	

Self-supervised learning: Generative

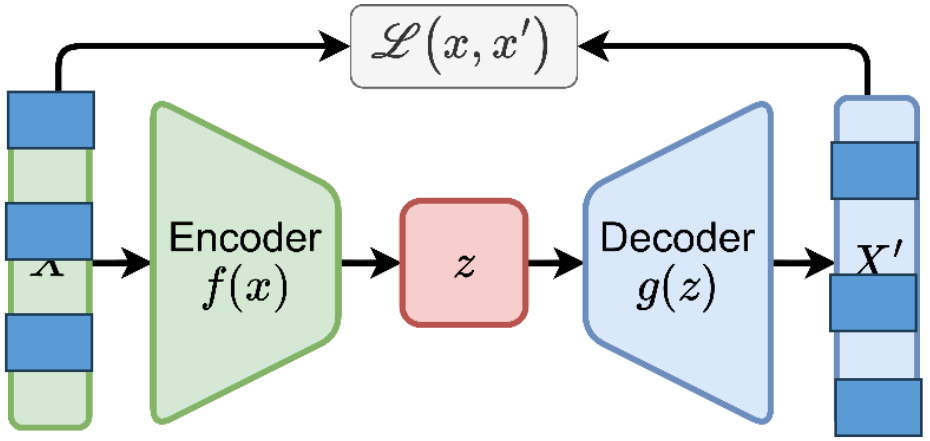
Masked autoencoder (wo/ compression)

- Apply masking to input and reconstruct it
- $A(x) = x_{input}, A^{-1}(x) = x_{target}$
- $\underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(g_\theta(f_\theta(x_{input})), x_{target})$

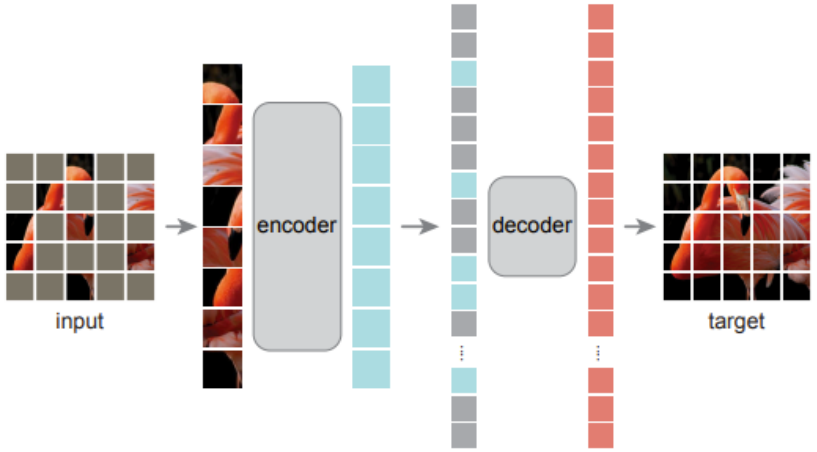
Masking strategies and amount:



Without Compression



Without Compression



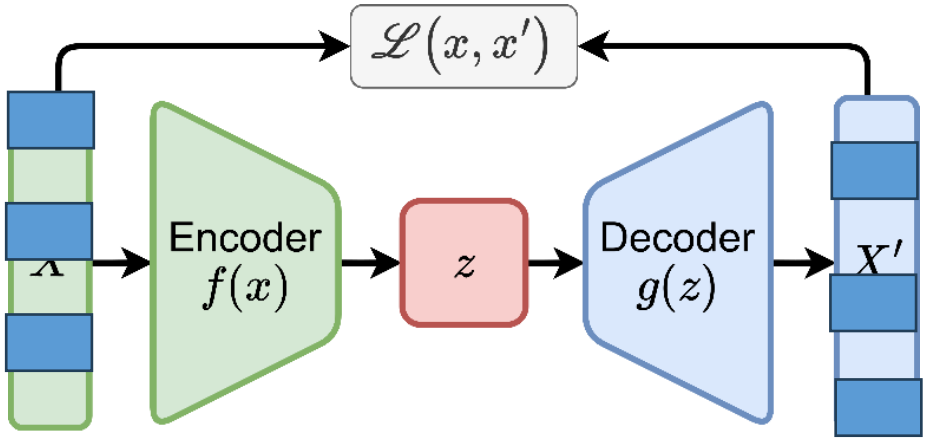
Self-supervised learning: Generative

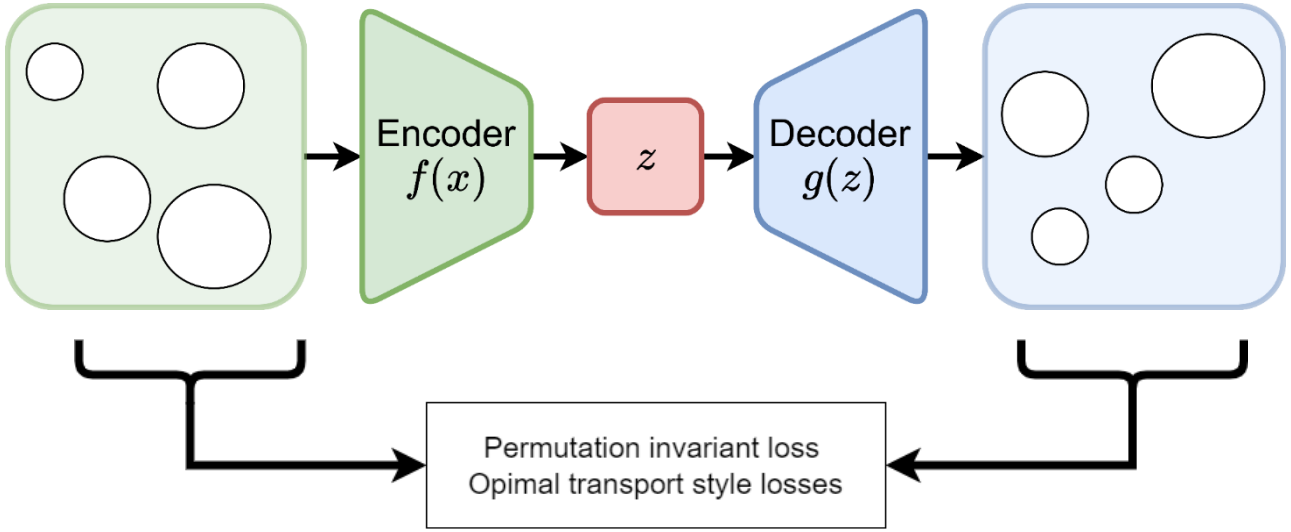
Masked autoencoder (wo/ compression)

- Apply masking to input and reconstruct it
- $A(x) = x_{input}, A^{-1}(x) = x_{target}$
- $\underset{\theta}{\operatorname{argmin}}\, \mathcal{L}(g_\theta(f_\theta(x_{input})), x_{target})$

Do you remember the issue with VAE?

Without Compression

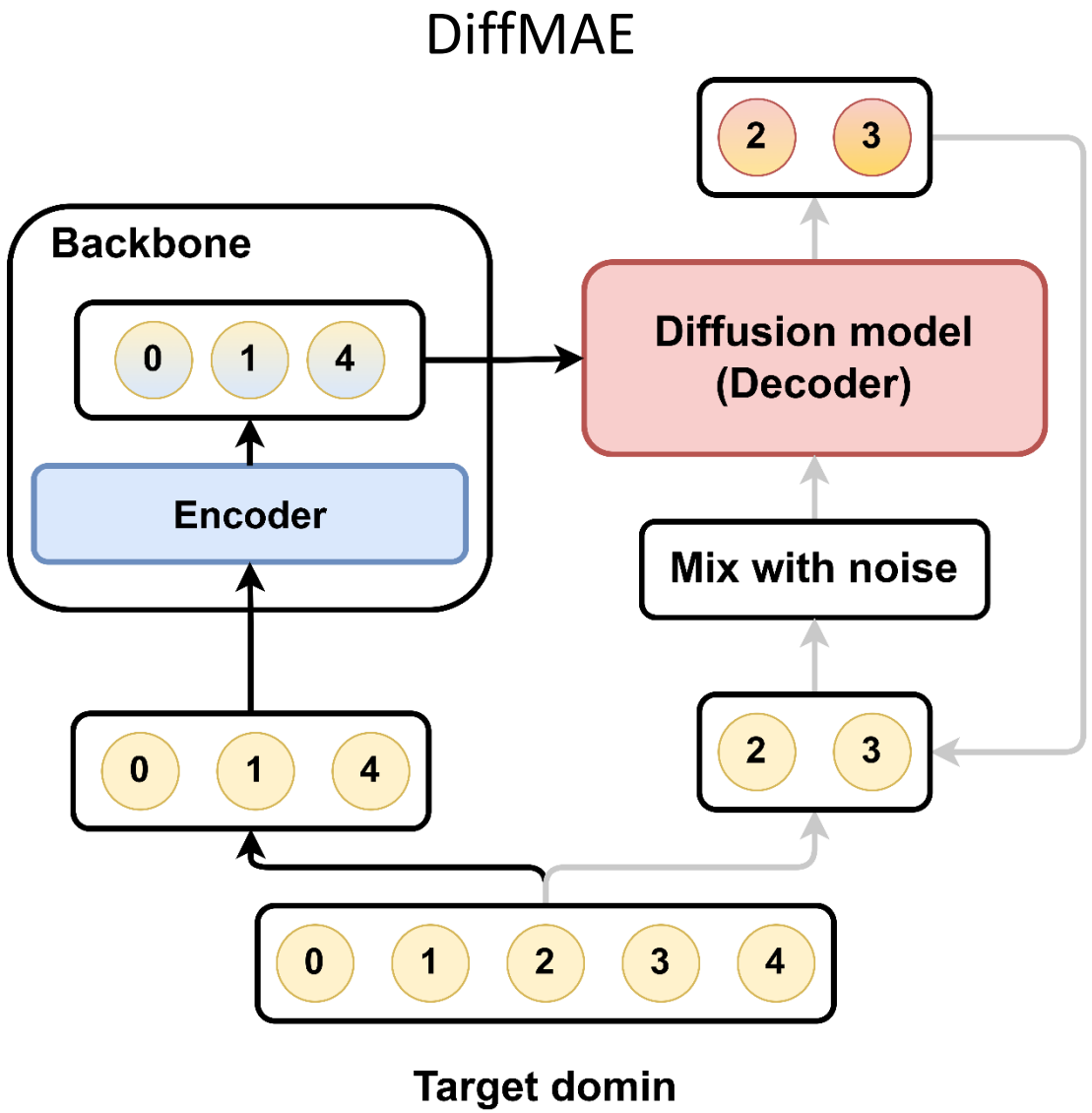




Self-supervised learning: Generative

Diffusion Models as Masked Autoencoders

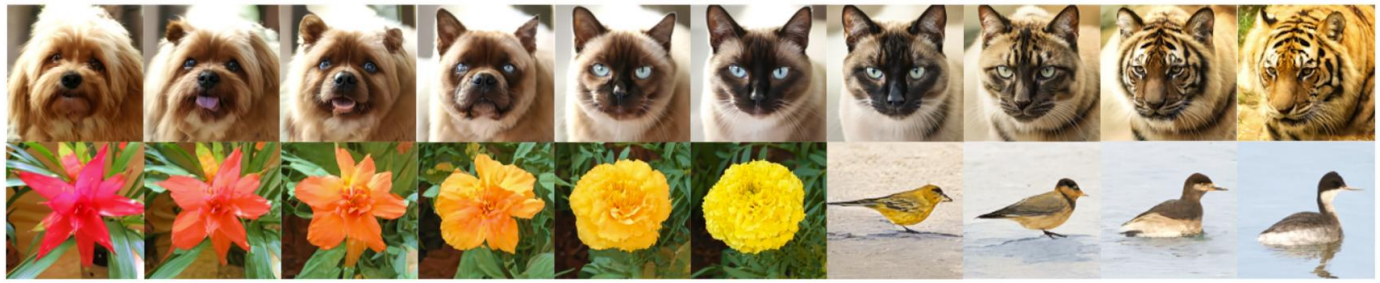
- Apply masking to input and reconstruct it
- $A(x) = x_{input}, A^{-1}(x) = x_{target}$
- $\underset{\theta}{\operatorname{argmin}} \mathcal{L}(g_\theta(f_\theta(x_{input})), x_{target})$
- Swap the decoder with a diffusion model
  - Same setup as MAE
  - Still no compression between $f_\theta(x_{input})$
  - But now compression can be introduced



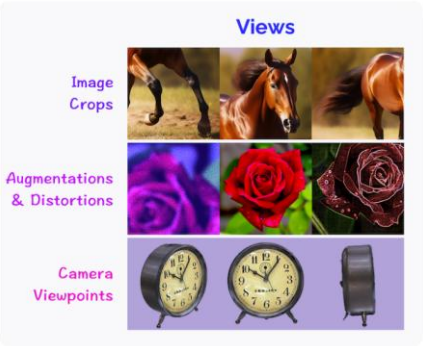
Self-supervised learning: Generative

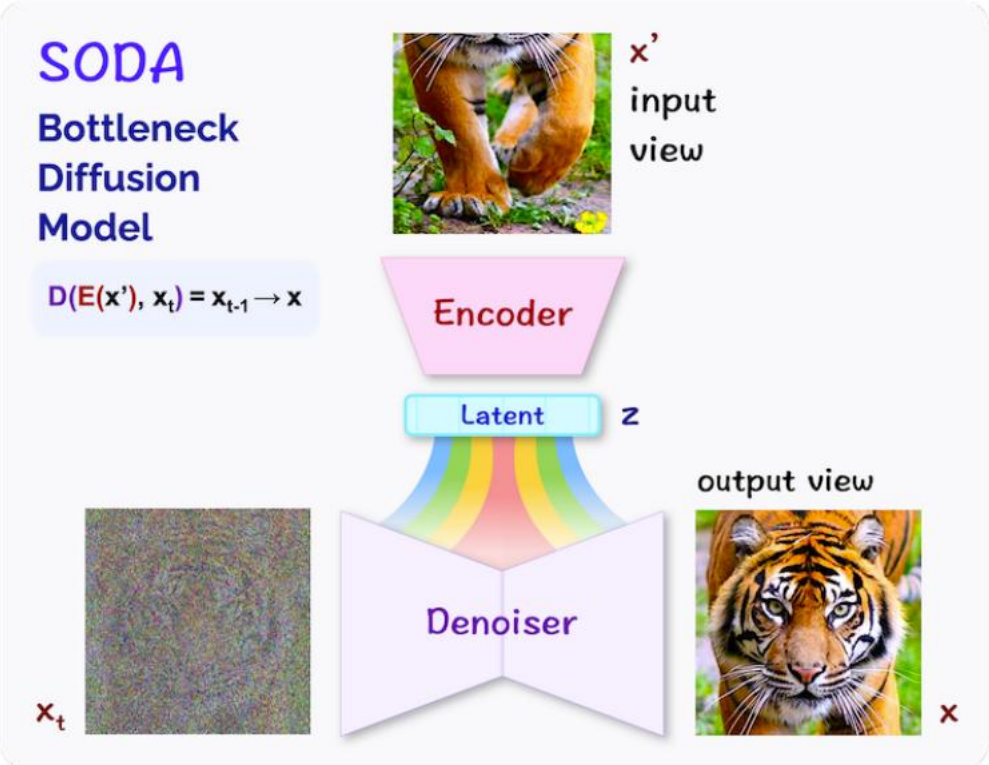
Bottleneck Diffusion Models for Representation Learning

- $A(x) = x_{input}, A^{-1}(x) = x_{target}$
  - A is different representations of the image
- Diffusion autoencoder
- Continuous latent representation
- Allow for interpolation in the latent



SODA





Self-supervised learning: Generative

Back to Masked AutoEncoders

- Drawbacks?
  - How do we ensure that the information we reconstruct is relevant?

MAE

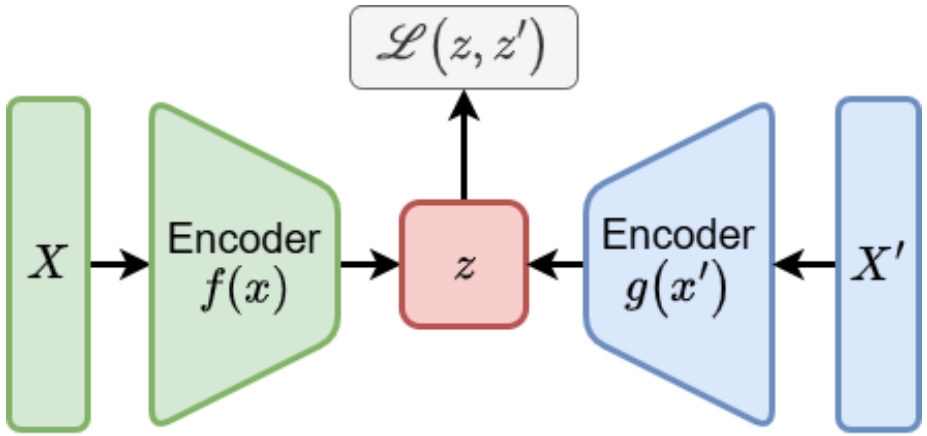
Apply masking

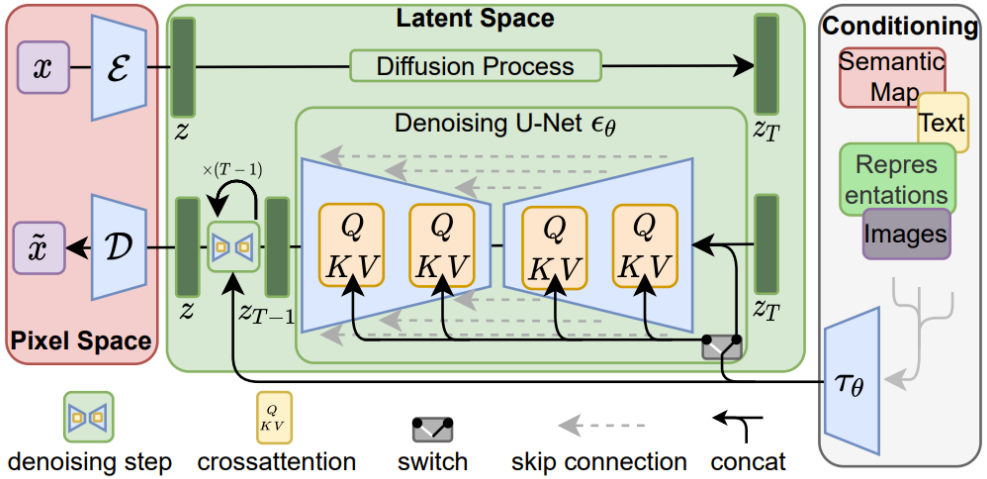
Malte Algren

Self-supervised learning: Generative

Back to Masked autoencoders

- Drawbacks?
  - How do we ensure that the information we reconstruct is relevant?
- For image generation (VQ)-VAE are used to compress image
  - Then generate the image in the latent space



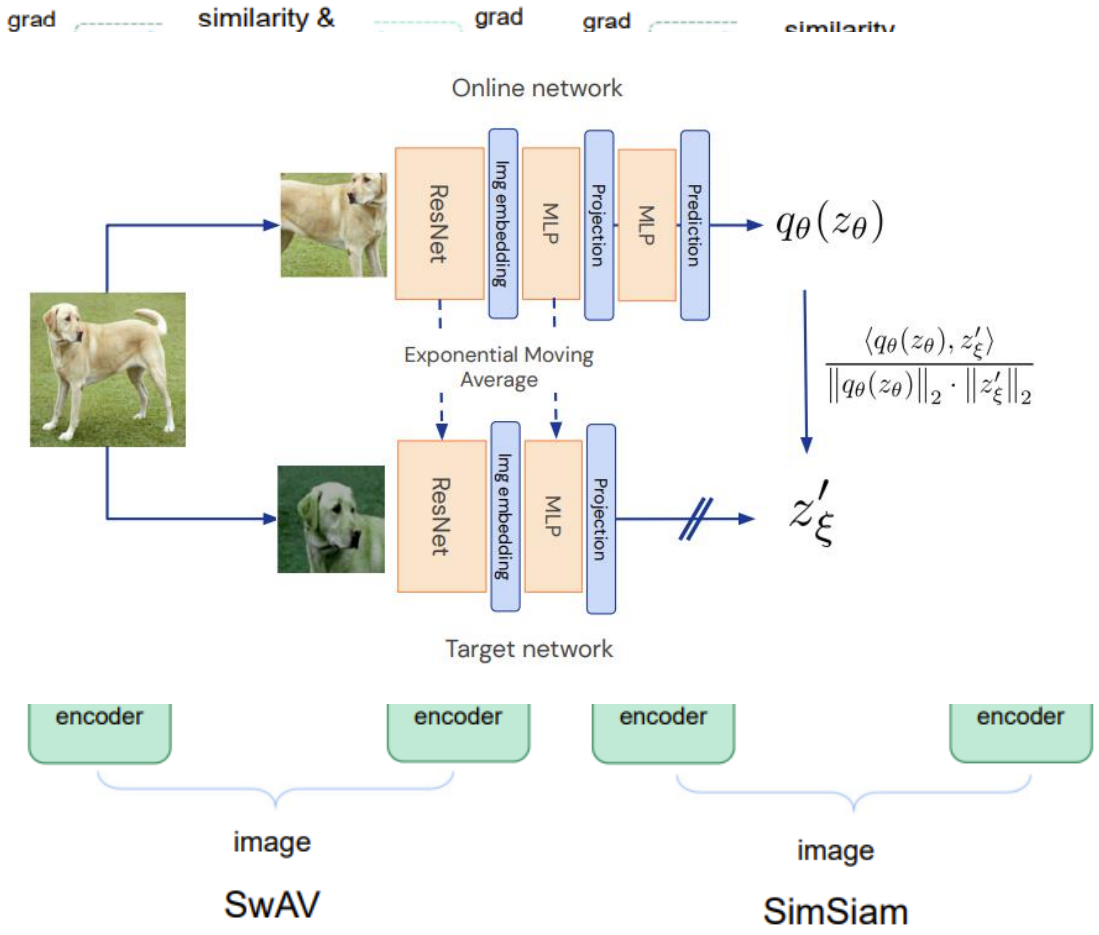


Self-supervised learning: Joint embedding

Compare features in the latent space

1. Augmentation in *x* to make x1 & x2
2. Encoder into latent space: z1 & z2
3. Similarity comparison

- Less redundant information in z1 & z2
- A lot of research in CL
  - SimCLR
  - BYOL
  - …



Self-supervised learning: Joint embedding

Compare features in the latent space

1. Augmentation in *x* to make x1 & x2
2. Encoder into latent space: z1 & z2
3. Similarity comparison

- Joint-Embedding Predictive Architecture

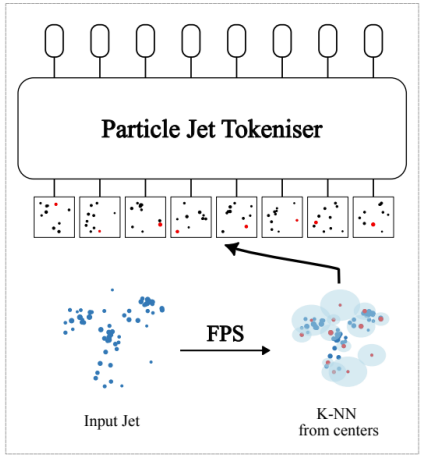
(a) **Joint-Embedding Architecture** (b) **Generative Architecture**

Self-supervised learning: Joint embedding

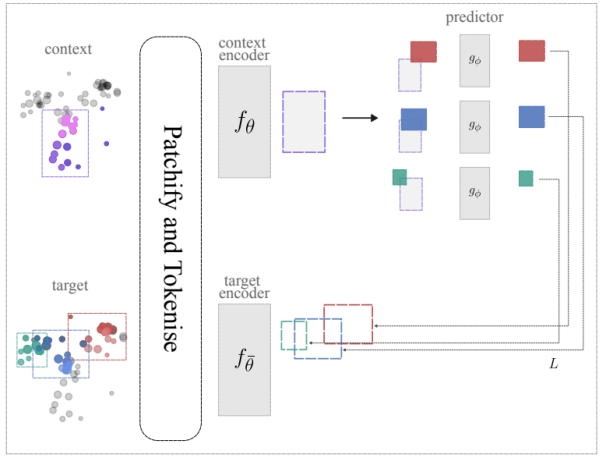
Compare features in the latent space

1. Augmentation in *x* to make x1 & x2
2. Encoder into latent space: z1 & z2
3. Similarity comparison

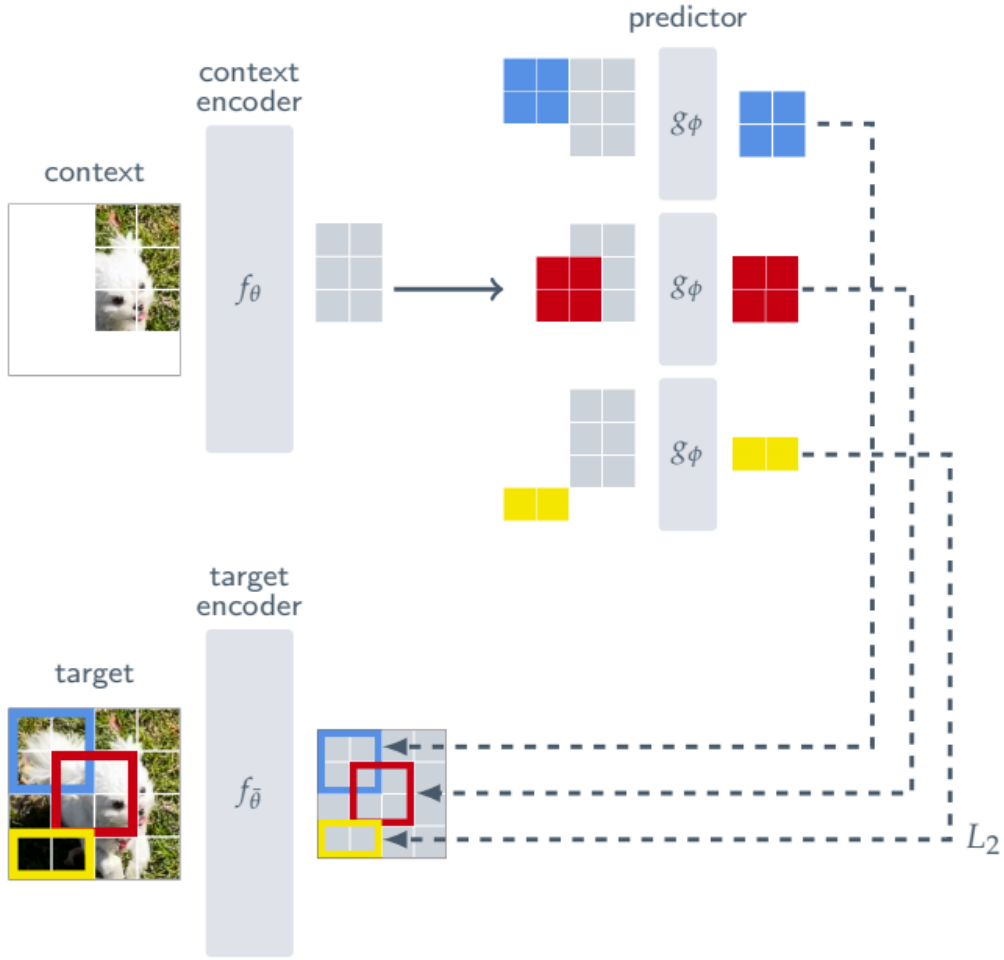
- Joint-Embedding Predictive Architecture
  - Extend to point clouds



Patchify and Tokenise



JEPA Framework



Self-supervised learning: Comparisons

Generative

1. Masked autoencoder MAE
2. Diffusion MAE

Properties

- Loss/reconstruction in *x*
- Work very well
- Stable
- Have been sort of left by the by CS

Joint embedding

1. SimCLR
2. BYOL
3. I-JEPA

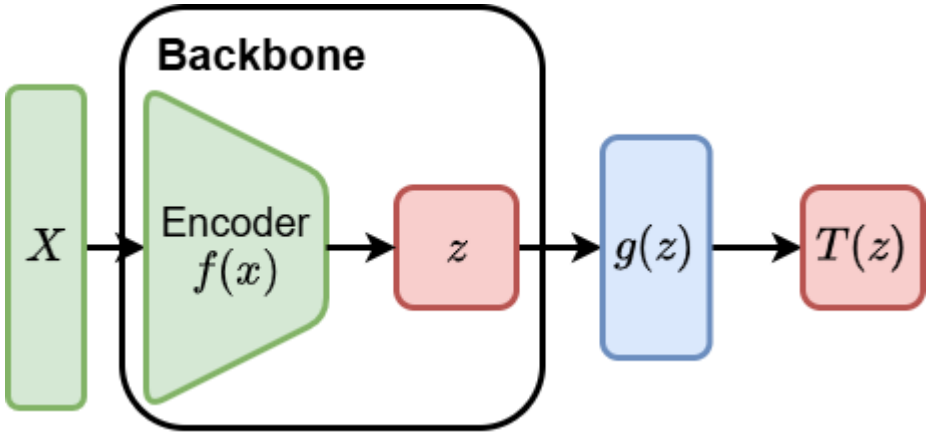
Properties

- Loss/reconstruction in *z*
- Very active when it comes to research
- Do also work well
- Can have model collapse

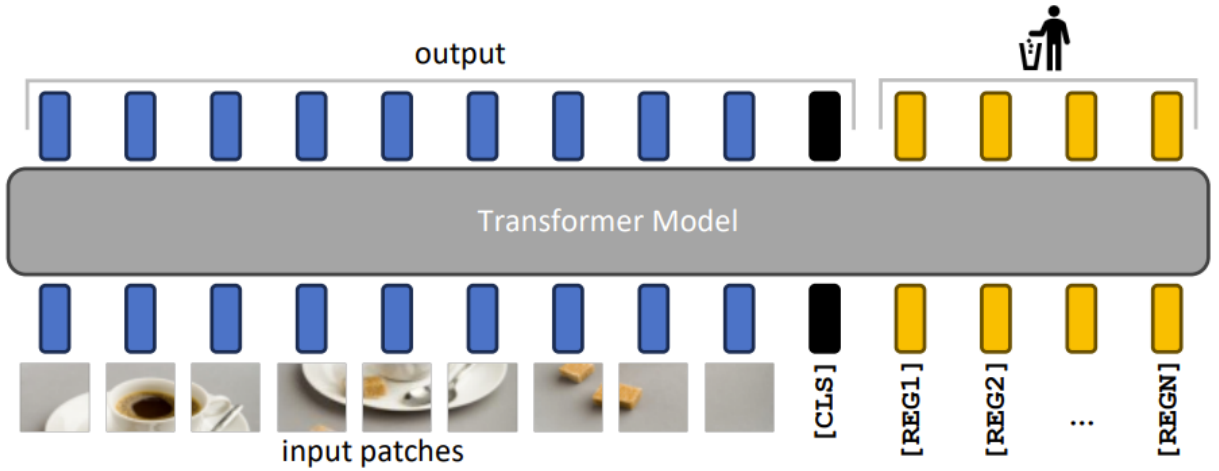
Quick points on finetuning

Now you have a pretrained model

- How to finetune it for a task?
  - You have access to $f(x) = z$



1. Size of model
   - Can you even backpropagate through the model?
2. Domain shift
   - ssl in data, finetune on simulation



Thanks for listening! Questions?