Apprentissage continu de représentations visuelles

ENSIMAG 2023-2024

Karteek Alahari & Diane Larlus

jeudi 7 décembre 2023

Organisation du cours

19/10/2023	11h15	12h45	ALAHARI Karteek	Intro + cours
26/10/2023	09h45	12h45	LARLUS Diane	Cours
09/11/2023	11h15	12h45	LARLUS Diane	**Articles 1 & 2**
16/11/2023	11h15	12h45	ALAHARI Karteek	Cours + **Article 3**
23/11/2023	11h15	12h45	LARLUS Diane	Cours
07/12/2023	11h15	12h45	LARLUS Diane	Cours
14/12/2023	09h45	12h45	LARLUS Diane + ALAHARI Karteek	**Articles 4, 5, 6** + Cours
21/12/2023	11h15	12h45	ALAHARI Karteek	Cours + **Article 7**
11/01/2024	11h15	12h45	ALAHARI Karteek	Cours + **Article 8**
18/01/2024	11h15	12h45	ALAHARI Karteek	Cours + Révisions

Apprentissage auto-supervisé: rappels

Apprentissage continu de représentations visuelles

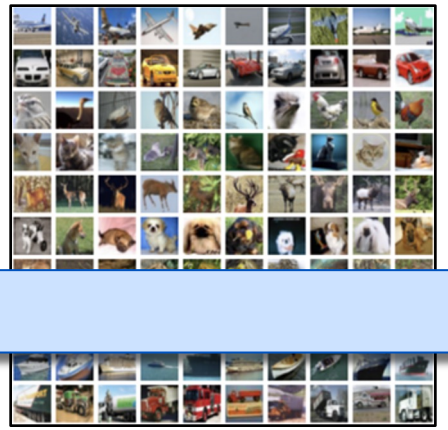
2023-2024

Fully-Supervised Classification

Images + labels

Proxy task

IMAGENET



Reducing annotation cost

Model

Visual representations

Target tasks

Image Classification

Object Detection

Instance Segmentation

Image Retrieval

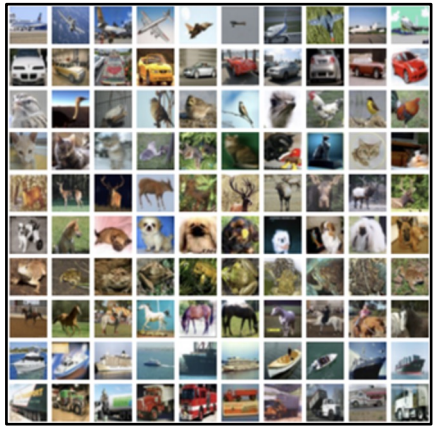
Learning transferable visual representations

No supervision

Reducing annotation cost

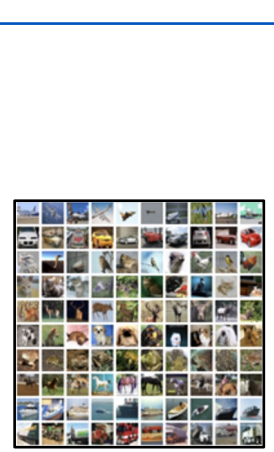
Fully-Supervised
fine-grained annotations
expert knowledge

Self-supervised
annotation-free images
no annotation required



~~labels~~

Apprentissage auto-supervisé: une première étape vers une tache plus concrête



IMAGENET

Images

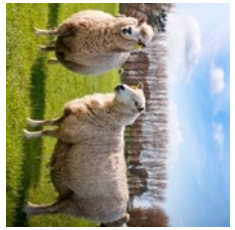
labels

Proxy task
= Rotation prediction

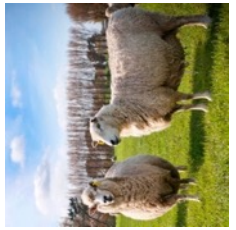
Encoder

Linear classifier

Class 1



Class 2



Class 3

Class 4

Target task = leaf classification

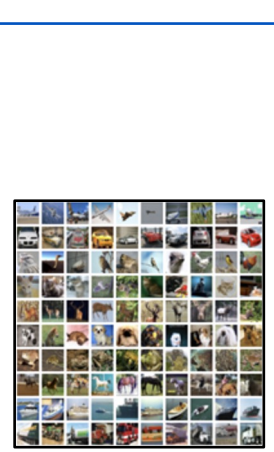
labels

Model

Linear classifier

Leaf class 1
Leaf class 2
…

Apprentissage auto-supervisé: une première étape vers une tache plus concrête



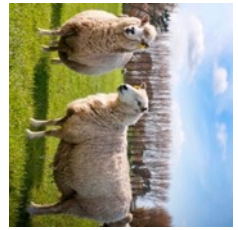
IMAGENET

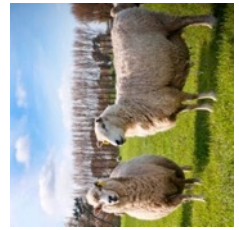
Images

Proxy task
= Rotation prediction

Encoder

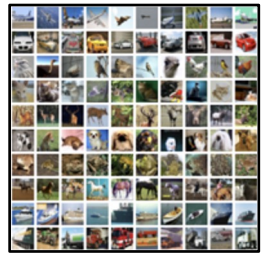
Linear classifier





labels

Target task = ImageNet classification

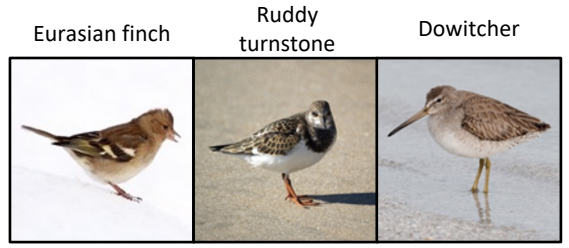


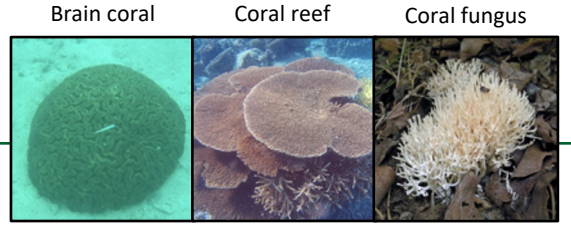
IMAGENET

labels

Model

Linear classifier





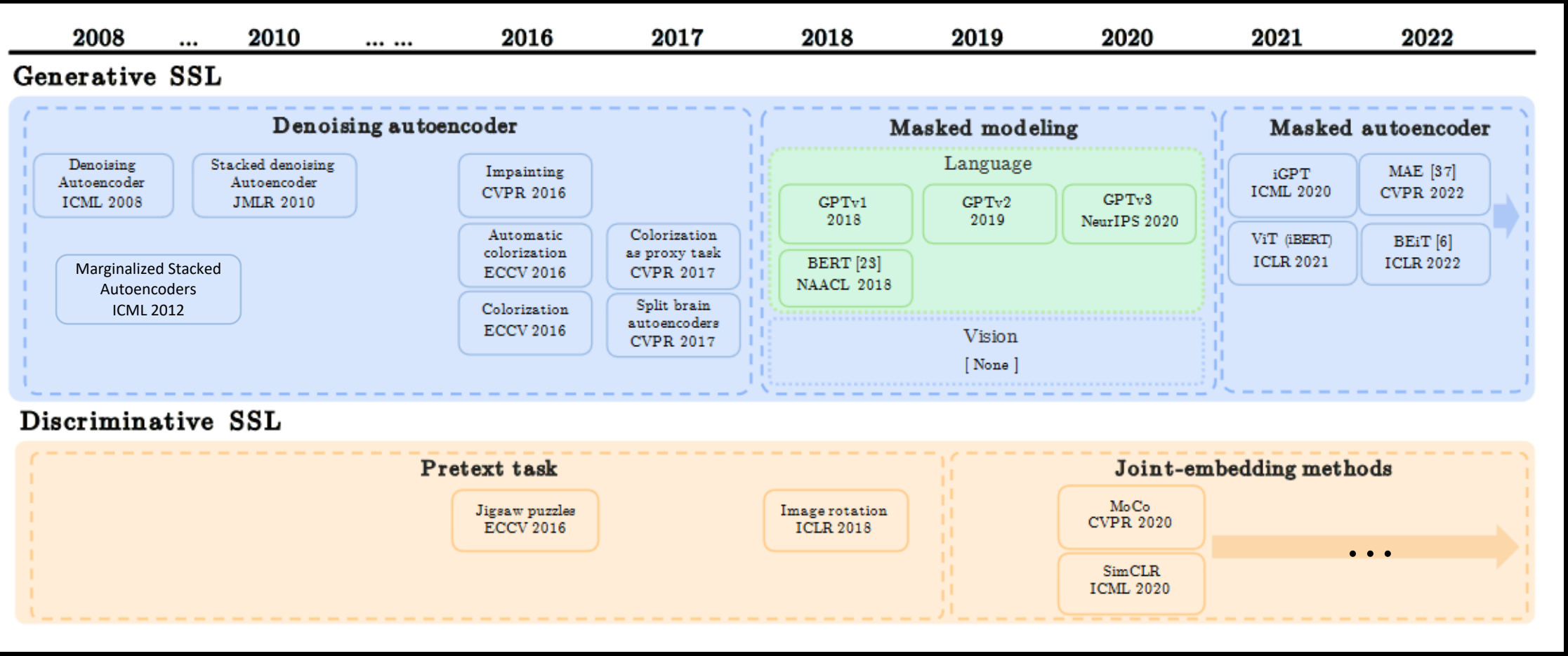
....

Tour d’horizon des méthodes d’apprentissage auto-supervisées (suite)

Apprentissage continu de représentations visuelles

2023-2024

TIMELINE OF VISUAL SSL



Zhang et al, A Survey on Masked Autoencoder for Self-supervised Learning in Vision and Beyond, ArXiv'22

[Slide: Gabriela Csurka]

Les méthodes discriminatives
Résumé – du dernier cours

Tâche prétexte: apprendre un modèle à résoudre une tâche de prédiction telle que de la classification, la discrimination d'instances, etc.

Notes:

- Apprentissage contrastif: tâche prétexte = discrimination d'instances / invariance à des transformations
  - Importance des échantillons négatifs
    - Utliser de plus grands batchs: SimCLR
    - Utiliser une mémoire en plus du contenu du batch: MoCo
    - Créer des négatifs synthétiques à la volée: MoChi
  - Méthodes sans négatifs: fonctions de coût spécifques: BarlowTwins ou méthodes d'auto-distilation: BYOL
- Lien entre l'apprentissage contrastif et le *clustering*. Tâche prétexte = classification, pour prédire des pseudo-labels: DeepCluster
  - L'étape de clustering est couteuse
    - Utilisation de l'auto-distillation: SwAV, DINO

Méthodes génératives

Apprentissage continu de représentations visuelles

2023-2024

DENOISING AUTOENCODERS

- Early work: image/feature corruption (random noise/drop-out)

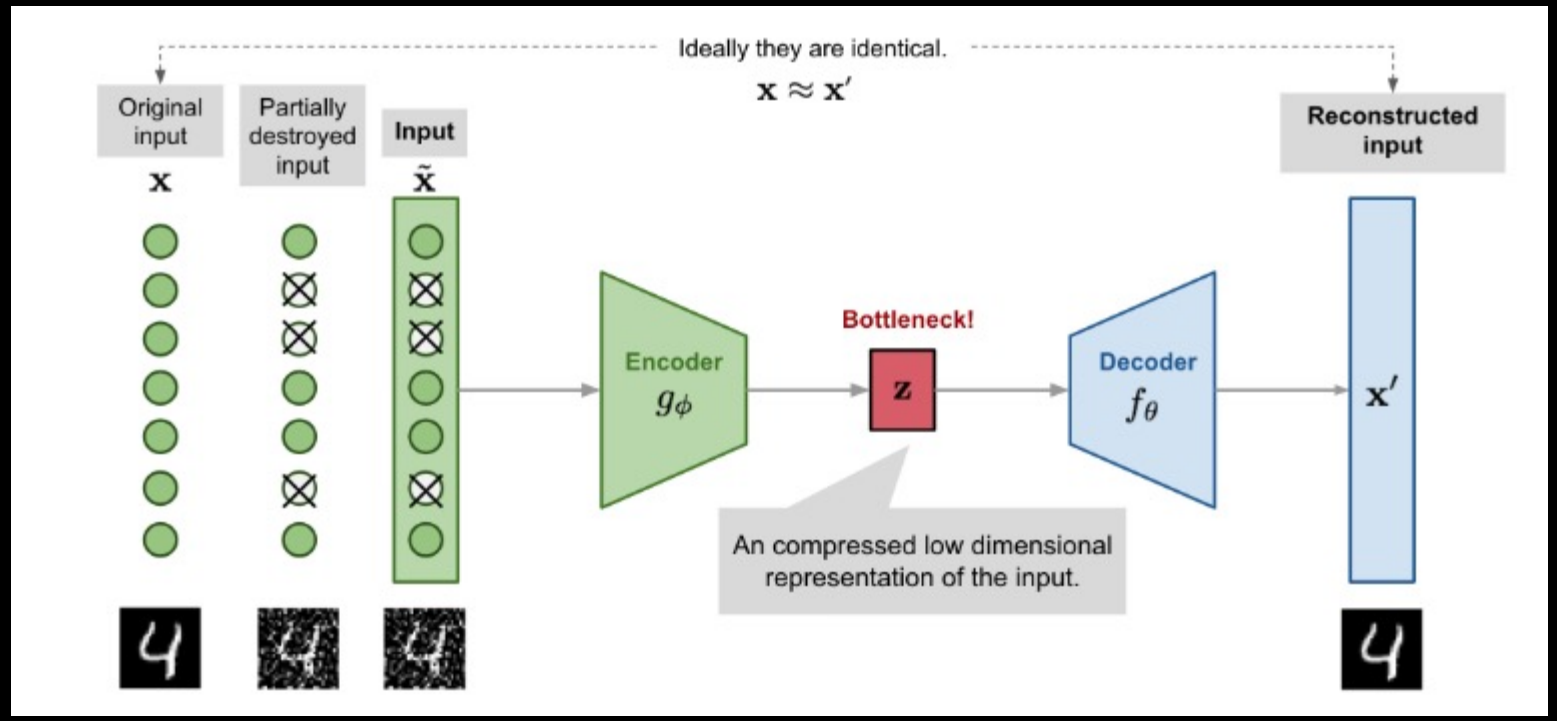


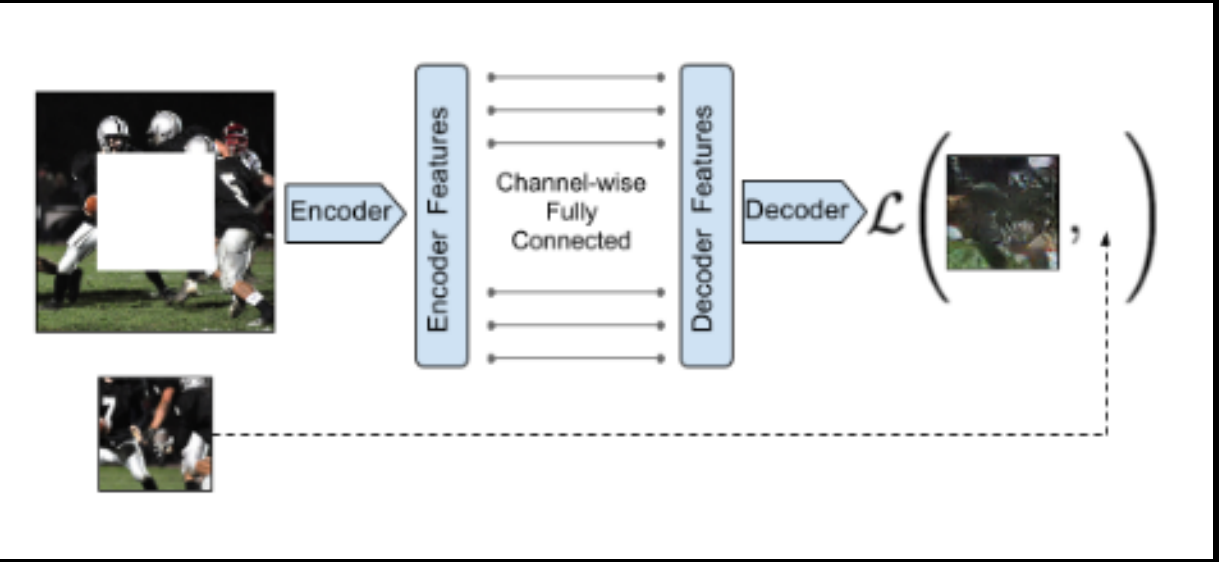
Image from Lilian Weng (https://lilianweng.github.io/posts/2018-08-12-vae/)

Vincent+@ICML'08, Vincent +@JMLR'10, Chen+@ICML'12

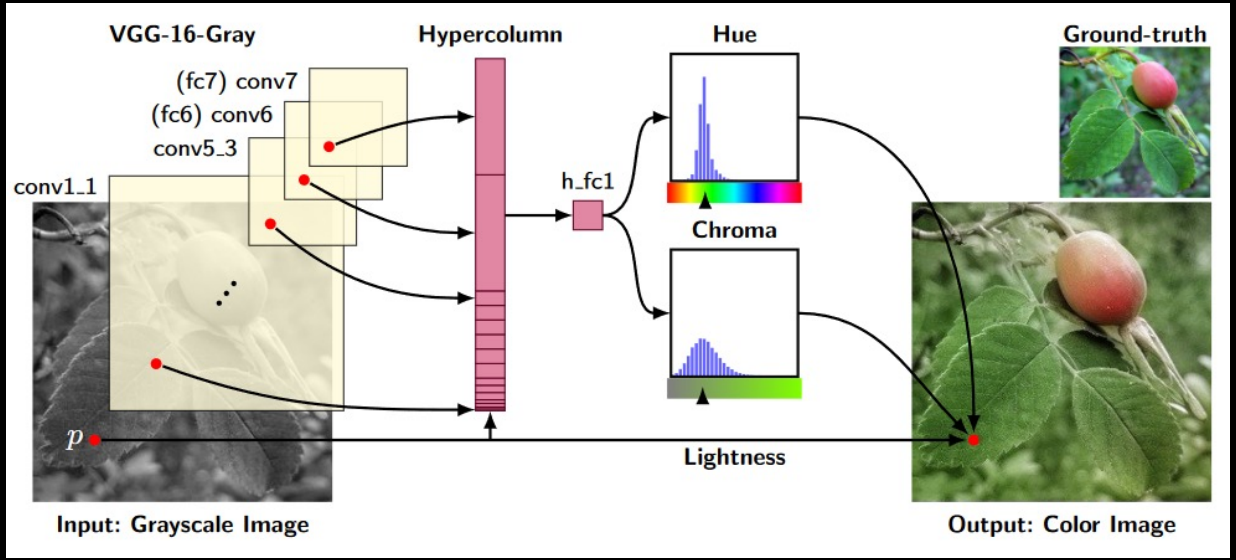
[Slide: Gabriela Csurka]

DENOISING AUTOENCODERS

Reconstruct image from general corruption (inpainting, colorization..)



Pathak+@CVPR'17



Larsson+@ECCV'17

Zhang@ECCV'16, Larsson+@ECCV'16, Pathak+@CVPR'17, Larsson+@CVPR'17

[Slide: Gabriela Csurka]

Les premières méthodes génératives
Résumé

Tâche prétexte: reconstruction d'images corrompues

Principe:

- Application de corruptions d'images plus ou moins sophistiquées à l'image de départ.
  Cela produit une image corrompue: une partie de l'information est manquante.
- Cette image corrompue est fournie en entrée du réseau / modèle à apprendre.
  Ce modèle est en général composé d'un encoder et d'un décodeur
  - L'encodeur produit une représentation de l'image.
  - Le décodeur regénère l'image à partir de la représentation obtenue de l'encodeur
- L'image initiale (non-corrompue) est utilisée comme vérité terrain pour l'apprentissage

Note: Le réseau de neurones appris a une première partie, l'encodeur, qui est le modèle qui nous intéresse vraiment et qui produit les représentations d'image. C'est cette partie qui va être réutilisée pour la ou les tâches cibles.
Le reste (et en particulier le décodeur) doit être appris aussi car il est nécessaire pour résoudre la tâche prétexte, mais il n'a aucun intérêt en dehors de cette tâche prétexte.
Cette partie du modèle est "défaussée" à la fin de l'apprentissage et n'est pas réutilisée.

Dans la suite: Les méthode génératives les plus récentes se sont inspirées de ce qui se fait en traitement naturel des langues.

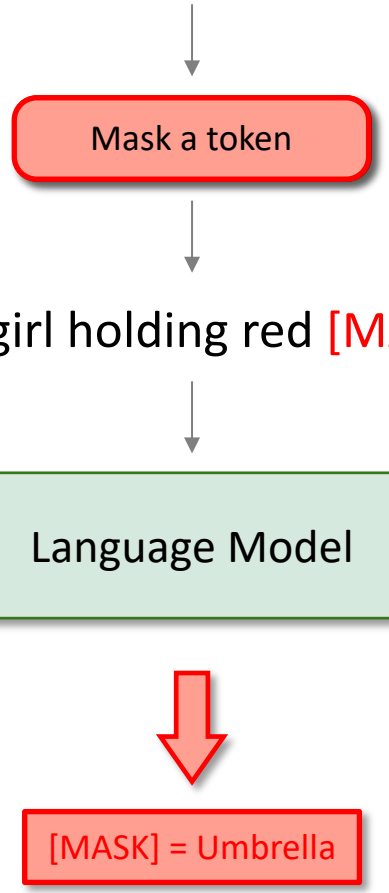
Learning transferable textual representations

Input:

BERT model
[Delvin *et al.* 2018]

Text

"Little girl holding red umbrella"

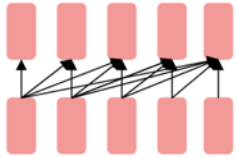


Pretext task:
Masked Language Modeling

Quelques notions de **traitement automatique des langues** (**TAL** ou ***NLP***)

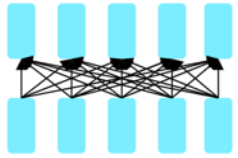
Source: *Natural Language Processing with Deep Learning* - CS224N/Ling284
Anna Goldie & John Hewitt – Sanford University

The neural architecture influences the type of pretraining, and natural use cases.



Decoders

- Language models!
- Nice to generate from; can't condition on future words
- **Examples:** GPT-2, GPT-3, LaMDA



Encoders

- Gets bidirectional context – can condition on future!
- Wait, how do we pretrain them?
- **Examples:** BERT and its many variants, e.g. RoBERTa

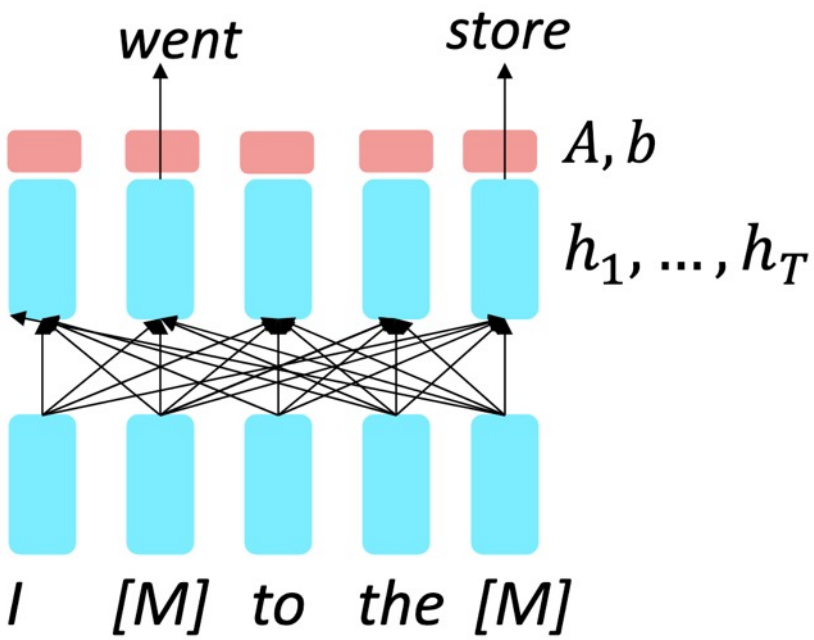
[Image credit: Anna Goldie & Jon Hewitt – CS224N/Ling284]

Pretraining encoders: what pretraining objective to use?

So far, we've looked at language model pretraining. But **encoders get bidirectional context,** so we can't do language modeling!

Idea: replace some fraction of words in the input with a special [MASK] token; predict these words.

Only add loss terms from words that are "masked out." If $\tilde{x}$ is the masked version of x, we're learning $p_\theta(x|\tilde{x})$. Called **Masked LM**.



[Devlin et al., 2018]

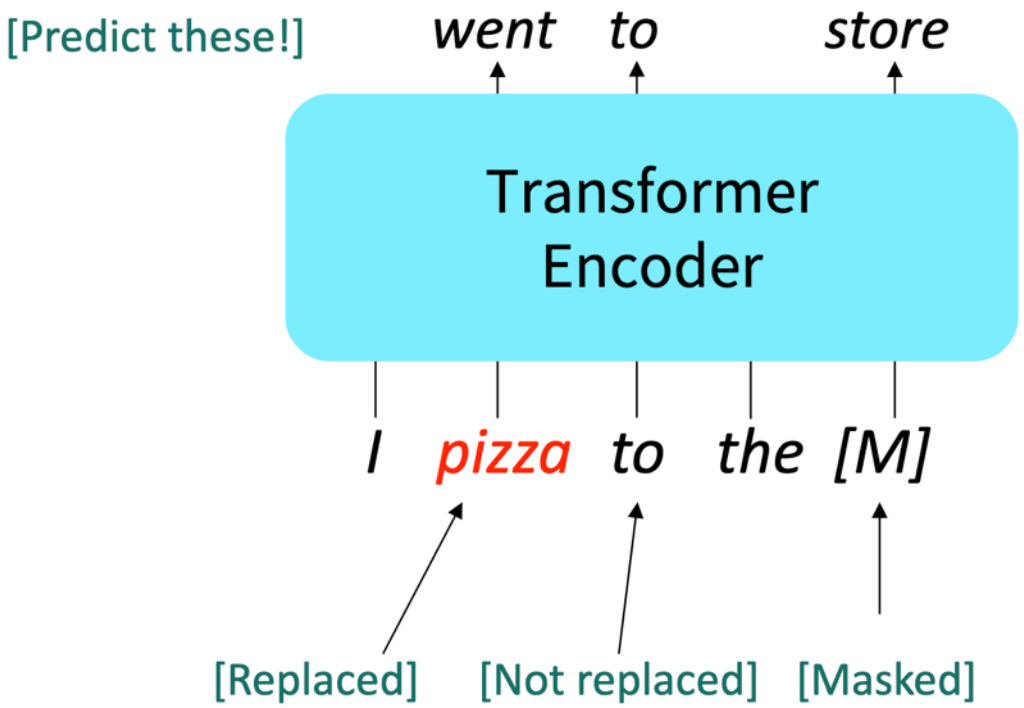
[Slide credit: Anna Goldie & Jon Hewitt – CS224N/Ling284]

BERT: Bidirectional Encoder Representations from Transformers

Devlin et al., 2018 proposed the "Masked LM" objective, open-sourced their model as the tensor2tensor library, and **released the weights of their pretrained Transformer (BERT).**

Some more details about Masked LM for BERT:

- Predict a random 15% of (sub)word tokens.
  - Replace input word with [MASK] 80% of the time
  - Replace input word with a random token 10% of the time
  - Leave input word unchanged 10% of the time (but still predict it!)
- Why? Doesn't let the model get complacent and not build strong representations of non-masked words. (No masks are seen at fine-tuning time!)

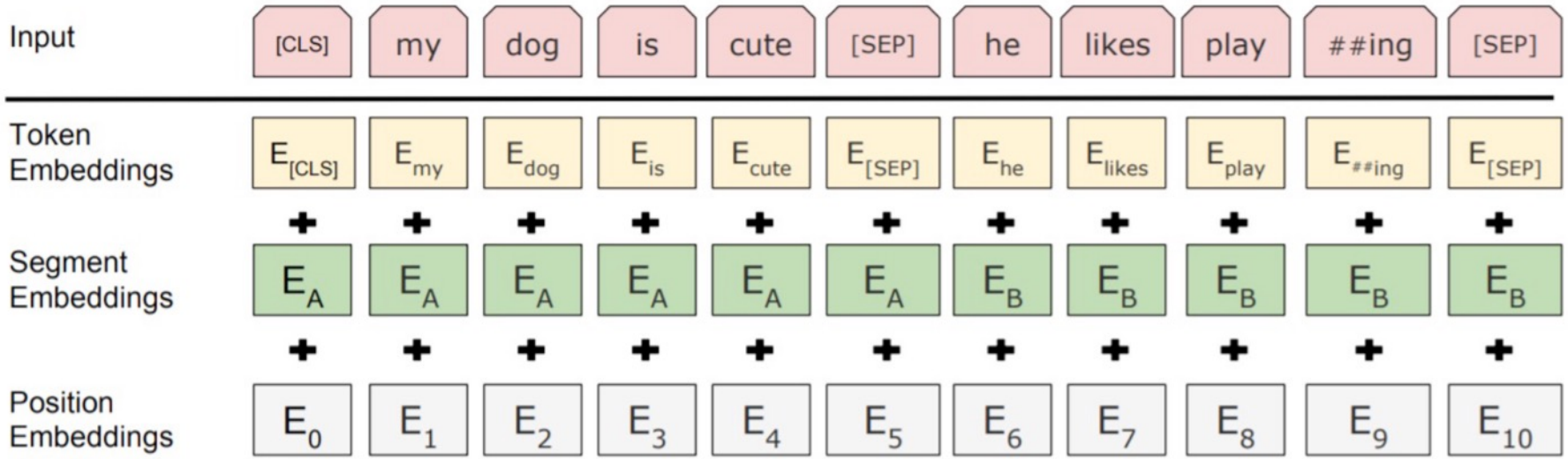


[Devlin et al., 2018]

[Slide credit: Anna Goldie & Jon Hewitt – CS224N/Ling284]

BERT: Bidirectional Encoder Representations from Transformers

- The pretraining input to BERT was two separate contiguous chunks of text:



- BERT was trained to predict whether one chunk follows the other or is randomly sampled.
  - Later work has argued this "next sentence prediction" is not necessary.

[Slide credit: Anna Goldie & Jon Hewitt – CS224N/Ling284]

[Devlin et al., 2018, Liu et al., 2019]

BERT: Bidirectional Encoder Representations from Transformers

Details about BERT

- Two models were released:
  - BERT-base: 12 layers, 768-dim hidden states, 12 attention heads, 110 million params.
  - BERT-large: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million params.
- Trained on:
  - BooksCorpus (800 million words)
  - English Wikipedia (2,500 million words)
- Pretraining is expensive and impractical on a single GPU.
  - BERT was pretrained with 64 TPU chips for a total of 4 days.
  - (TPUs are special tensor operation acceleration hardware)
- Finetuning is practical and common on a single GPU
  - "Pretrain once, finetune many times."

[Slide credit: Anna Goldie & Jon Hewitt – CS224N/Ling284]

[Devlin et al., 2018]

MASKED IMAGE MODELLING (MIM)

Inspired by masked language modelling,
mask image regions and reconstruct them from the context.
Visual transformer-based architectures (most often ViT)

Tokenizer-based MIM (iGPT, iBERT, BEiT, mc-BEiT, CAE, MAGE, PeCo)

- relying on a pretrained visual dictionary generated with discrete variational autoencoder (dVAE) trained using ImageNet or Dall-E
- reconstruct the target tokens (instead of the image pixels)

End-to-end masked autoencoder (MAE, SimMIM, UM-MAE, CIM)

- single step without explicit visual vocabulary
- reconstruct the masked/corrupted image

[Slide: Gabriela Csurka]

Les méthodes génératives les plus récentes

Tâche prétexte: reconstruction d'images corrompues (toujours)

Deux types d'approches:

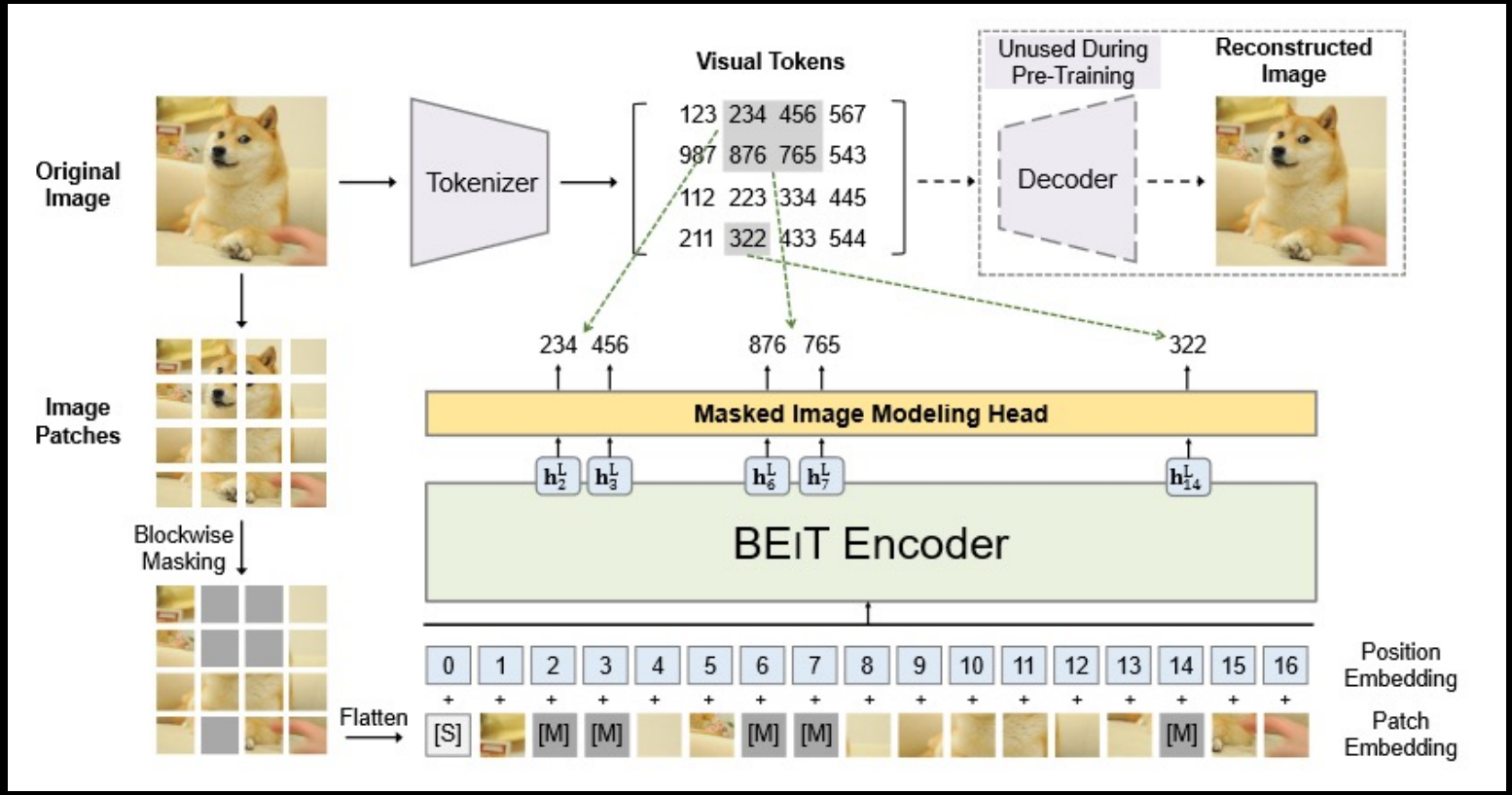
- Certaines méthodes simplifient le problème, et reconstruisent l'image uniquement à travers les "tokens" d'une image. On suppose l'existence d'un vocabularie visuel capable de transformer une image en un ensemble de "tokens". Les modèles apprennent seulement à reconstruire les "tokens" de l'image d'entrée.
- D'autres méthodes reconstruisent directement les pixels des images.

Voyons tout d'abord un exemple de ***Tokenizer-based Mask Image Modeling.***

BERT PRE-TRAINING OF IMAGE TRANSFORMERS (BEiT)

Reconstruction based on a visual vocabulary (pre-trained with dVAE)

Masked visual tokens predicted based on the encoding of the corrupted image



Bao+@ICLR'22

[Slide: Gabriela Csurka]

Les méthodes génératives les plus récentes
Résumé

Tâche prétexte: reconstruction d'images corrompues (toujours)

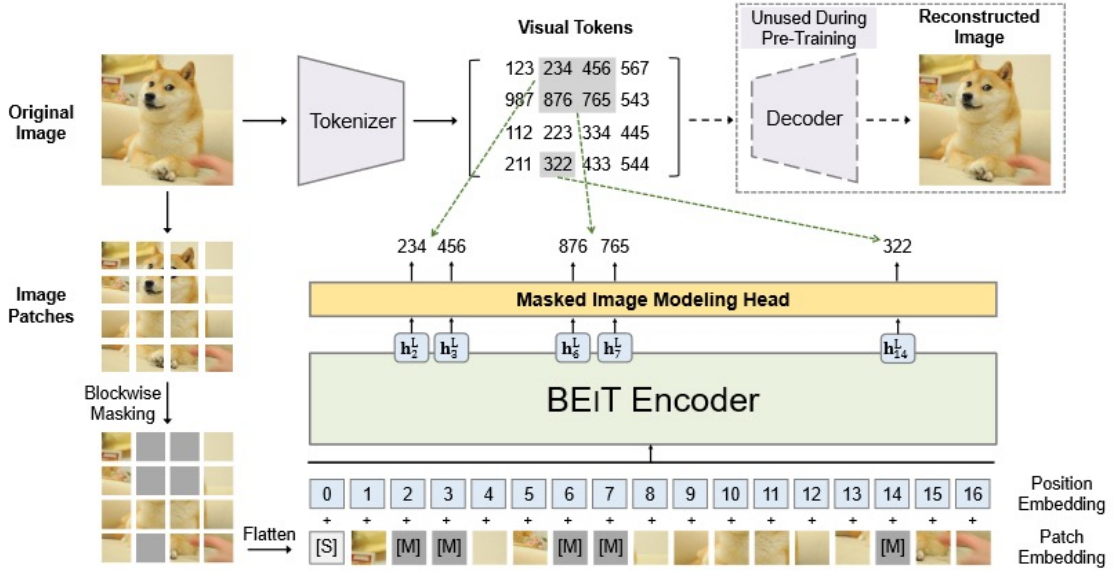
Une différence essentielle entre le texte et les images réside dans le fait que l'unité de base en texte est le mot, qui a un rôle sémantique, alors qu'en image le signal de base est le pixel, qui est moins informatif. Cela pose un problème réel pour la prédiction d'informations visuelles masquées.

Une solution possible est d'imiter le vocabulaire utilisé en texte par un vocabulaire visuel qui discrétise les imagettes (*patchs*) en tokens. C'est le choix fait par BEiT.

La tâche prétexte devient **la prédiction des *tokens* des parties masquées**.

Le processus est le suivant:

- Découper l'image en 4x4 imagettes (*patchs*)
- Appliquer un masquage aléatoire de ces imagettes
- Ajouter un encodage de la position (*positional encoding*)
- Fournir ces représentations à une architecture basée sur des transformeurs
- Obtenir une prédiction pour l'image, en déduire les « tokens » des parties masquées



Les méthodes génératives les plus récentes
Résumé

Tâche prétexte: reconstruction d'images corrompues (toujours)

Une différence essentielle entre le texte et les images réside dans le fait que l'unité de base en texte est le mot, qui a un rôle sémantique, alors qu'en image le signal de base est le pixel, qui est moins informatif. Cela pose un problème réel pour la prédiction d'informations visuelles masquées.

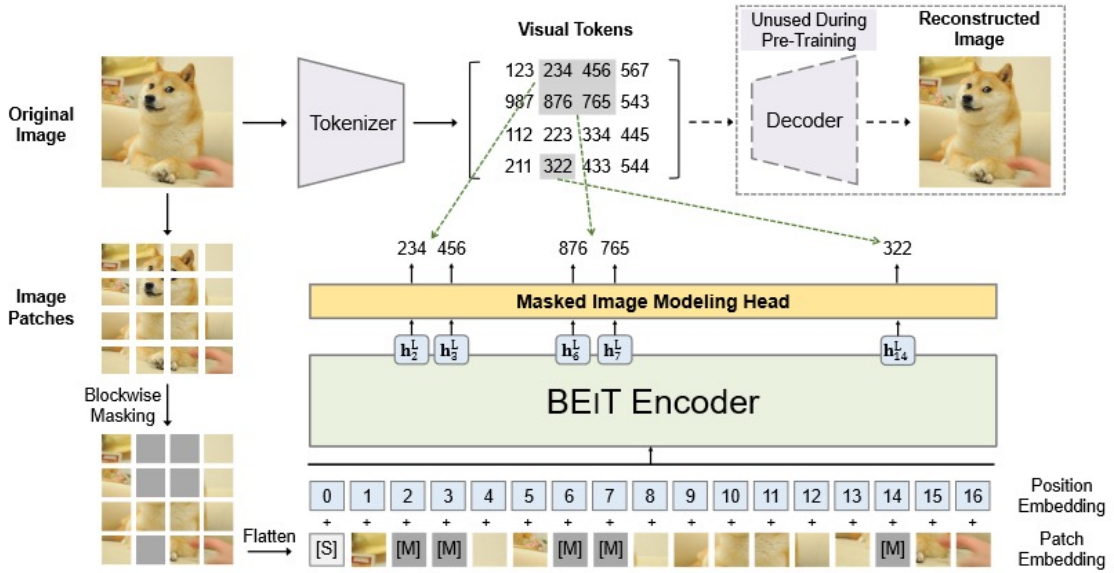
Une solution possible est d'imiter le vocabulaire utilisé en texte par un vocabulaire visuel qui discrétise les imagettes (*patchs*) en tokens. C'est le choix fait par BEiT.

La tâche prétexte devient **la prédiction des *tokens* des parties masquées**.

Comment obtenir ces « tokens visuels »?

Une solution est d'utiliser un « *image tokenizer* ».

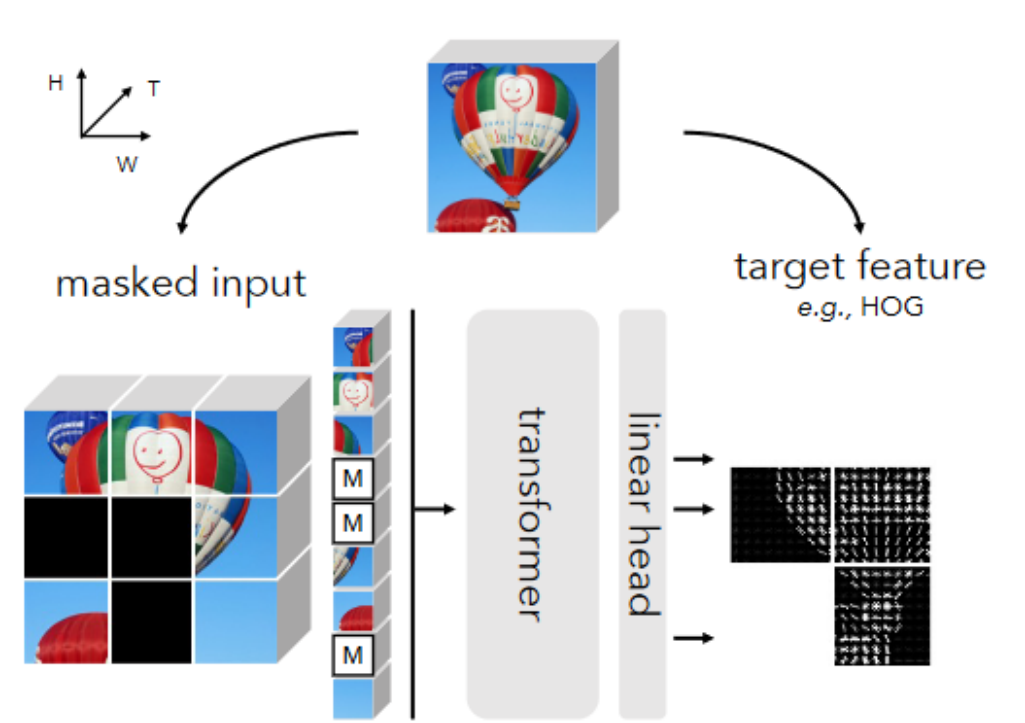
Pour cela, BEiT utilise un auto-encoder variationnel (*variational auto-encoder*), et plus précisément celui qui est entrainé dans le modèle DALL-E d'OpenAI.



MASKED FEATURE PREDICTION (MaskFeat)

Study five different types of target features :

- pixel colors, HOG, dVAE, deep features, pseudo-labels



feature type	one-stage	variant	top-1
scratch	-	MViT-S [56]	81.1
pixel	✓	RGB	80.7
image descriptor	✓	HOG [22]	**82.2**
dVAE	✗	DALL-E [73]	81.7
unsupervised feature	✗	DINO [9], ViT-B	**82.5**
supervised feature	✗	MViT-B [31]	81.9

MaskFeat with handcrafted HOG features achieve competitive performance, suggesting a target tokenizer generated by dVAE might be unnecessary.

Wei+@ArXiv'21

[Slide: Gabriela Csurka]

Les méthodes génératives les plus récentes

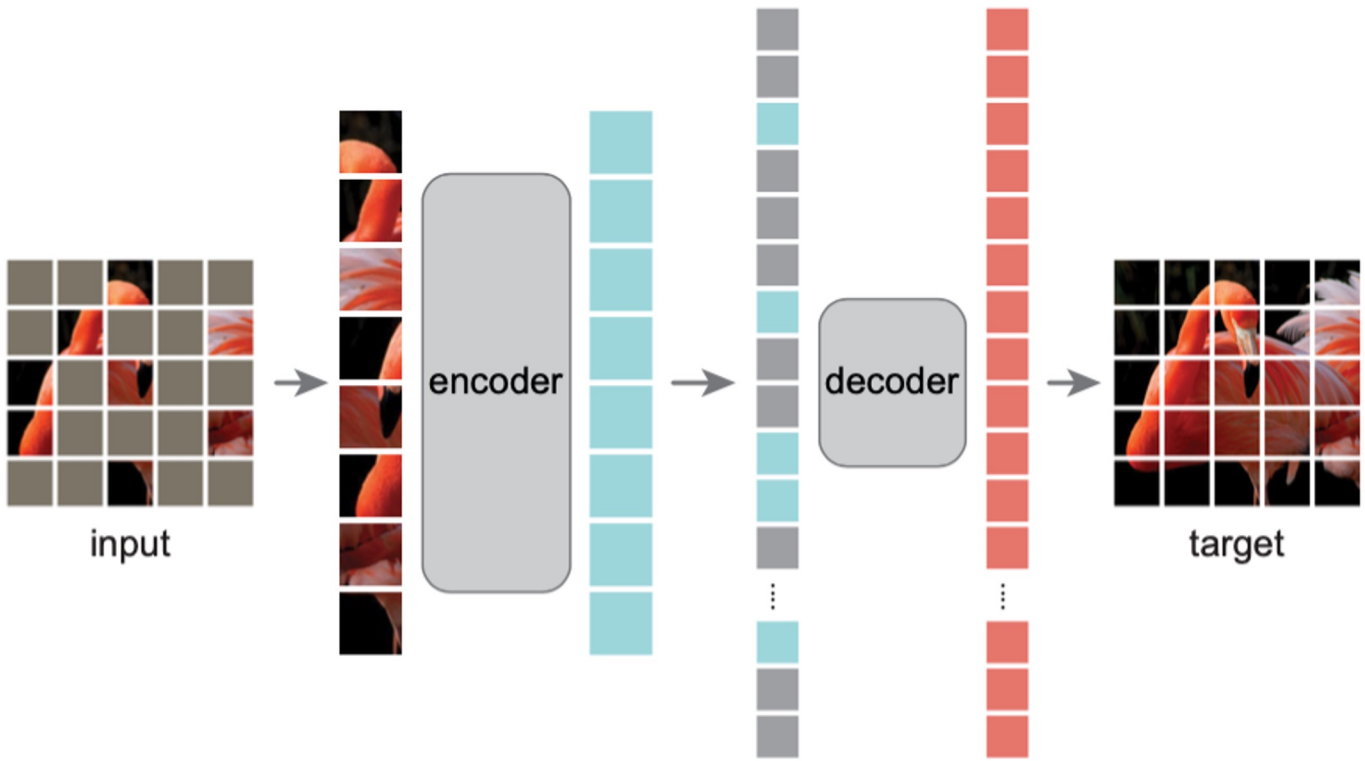
Voyons maintenant des exemples de ***End-to-end masked autoencoder***

Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)



Article 6

“what makes masked autoencoding different between vision and language?”

- **Architecture gap:** Until recently, architectures were different (CNN).
  - Solved: with Vision Transformers (ViTs) this is no longer the case
- **Different role for the decoder:**
  - In vision, we reconstruct pixels - output is of a lower semantic level than common tasks.
  - In language, the decoder predicts missing words that contain rich semantic information
  - *A tokenizer tries to solve this issue (see BeiT)*
- **Information density** is different between language and vision
  - Languages: highly semantic. Predicting only a few missing words appears to induce sophisticated language understanding
  - Images: natural signals with *heavy spatial redundancy*. A missing patch can be recovered from neighboring patches with little high-level understanding

[Slide credit: Yannis Kalantidis]

How well do humans do reconstruction?

Ground-Truth

[Slide credit: Yannis Kalantidis]

How well do humans do reconstruction?

Prediction

[Slide credit: Yannis Kalantidis]

How well do humans do reconstruction?

Ground-Truth

[Slide credit: Yannis Kalantidis]

How well do humans do reconstruction?

Prediction

[Slide credit: Yannis Kalantidis]

MAE does pretty well

Prediction

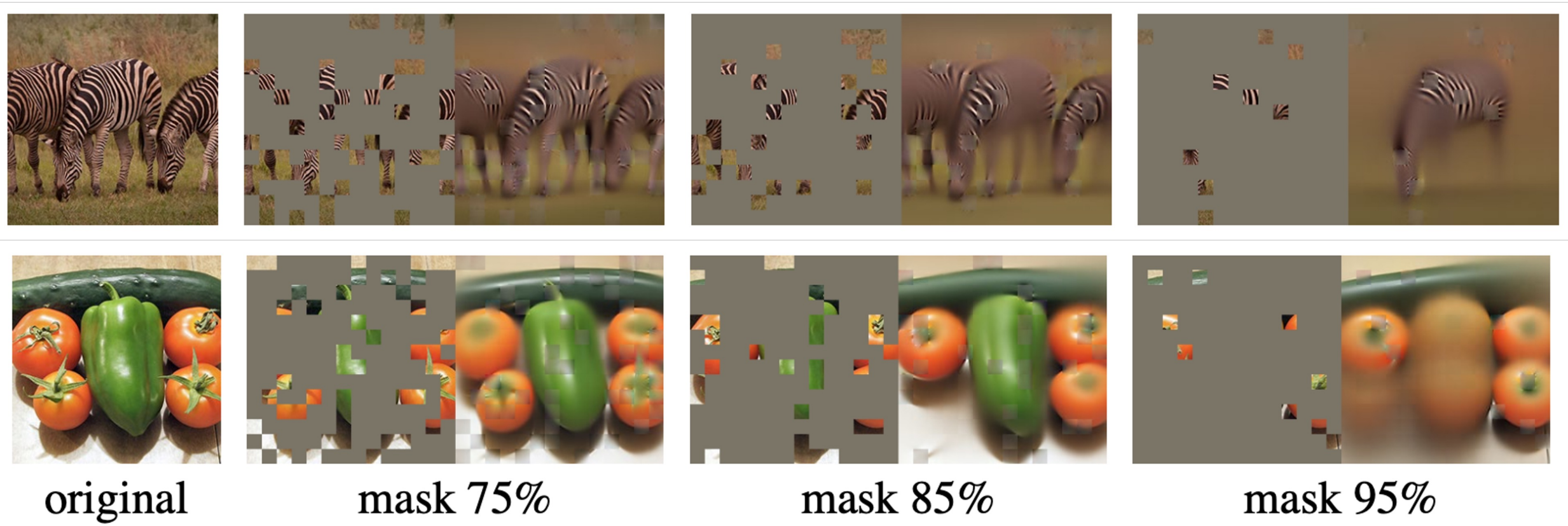
[Slide credit: Yannis Kalantidis]

MAE does pretty well

Ground-Truth

[Slide credit: Yannis Kalantidis]

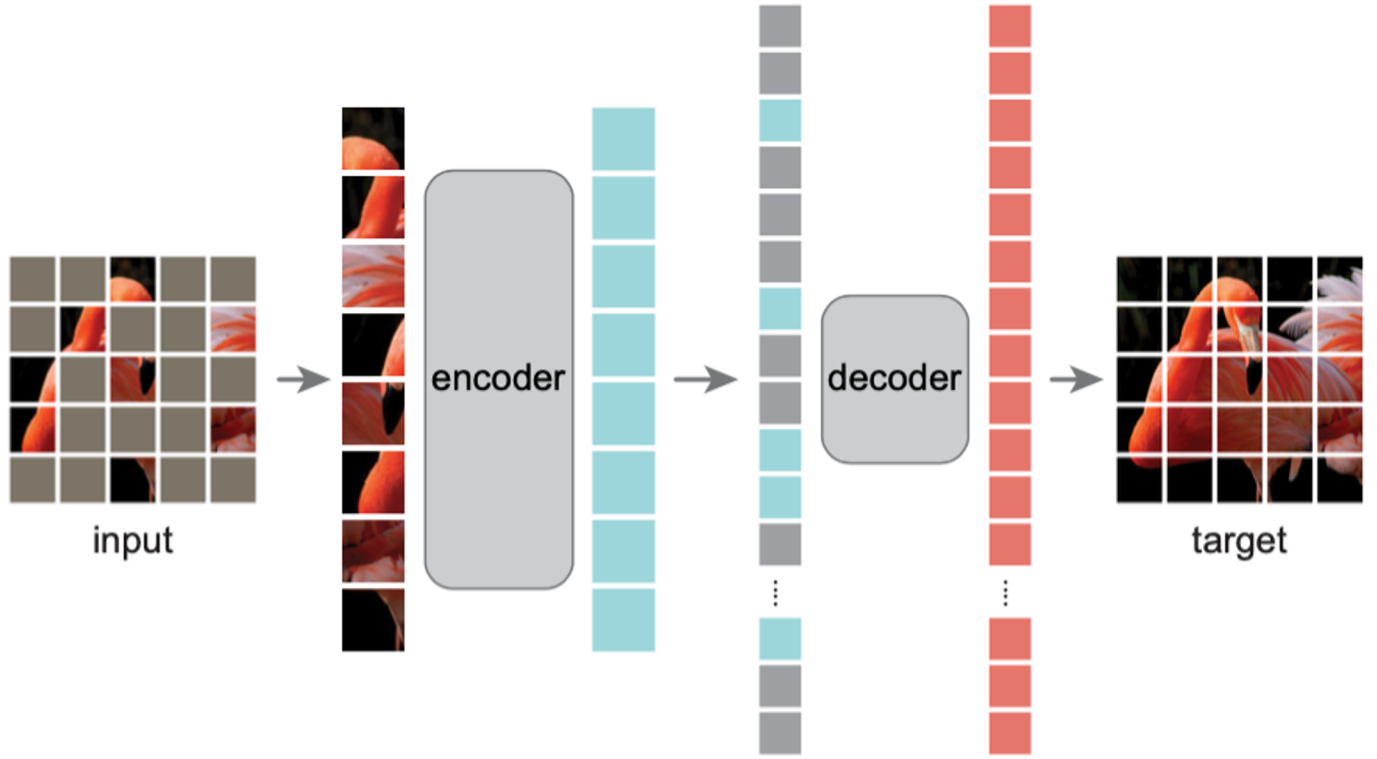
MAE does pretty well



[Slide credit: Yannis Kalantidis]

MAE – Main take home messages

- no tokenizer!
- Masked tokens introduced *after* the encoder
- In the end we keep the encoder only



[Slide credit: Yannis Kalantidis]

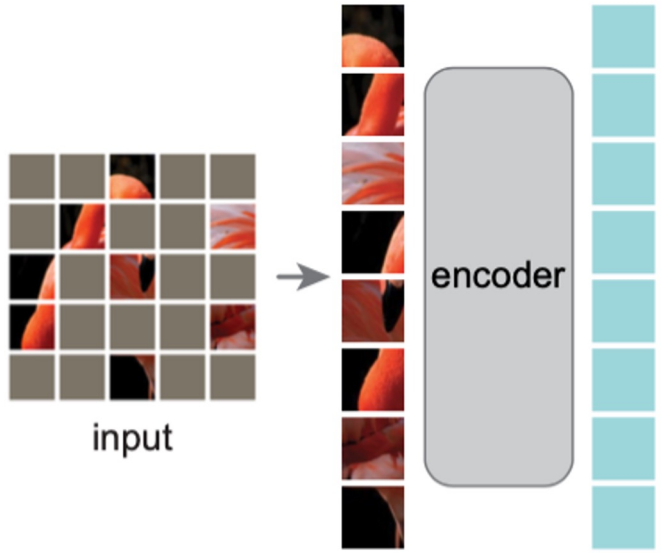
Masking

- Divide an image into regular non-overlapping patches.
- Sample a subset of patches and mask (i.e., remove) the remaining ones
- Sampling strategy: “random sampling” (without replacement, following a uniform distribution)
- Crucial: Eliminate the spatial redundancy:
  - Random sampling with **a high masking ratio**
  - This pretext task cannot easily be solved by extrapolation from visible neighboring patches

[Slide credit: Yannis Kalantidis]

MAE encoder

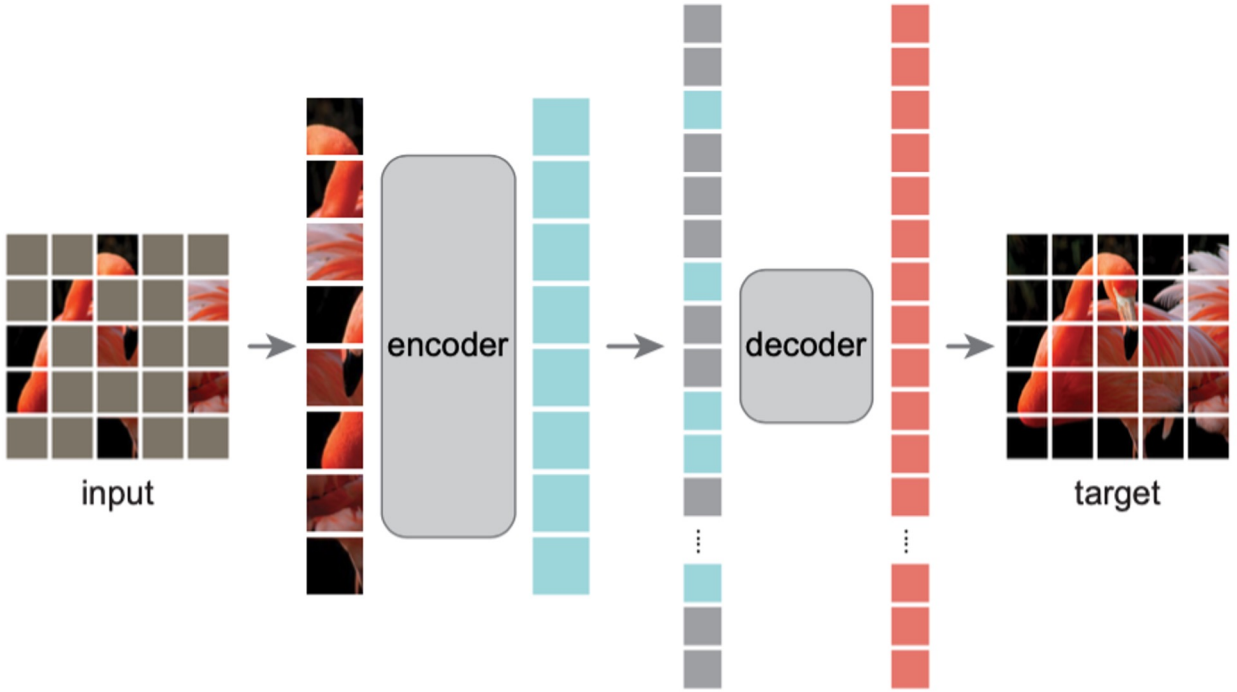
- a ViT but applied *only on unmasked patches*
- This trick makes the method very efficient:
  - The encoder only operates on a small subset (e.g., 25%) of the full set of patches.
  - This allows to train large encoders with only a fraction of the compute and of the memory



[Slide credit: Yannis Kalantidis]

MAE decoder

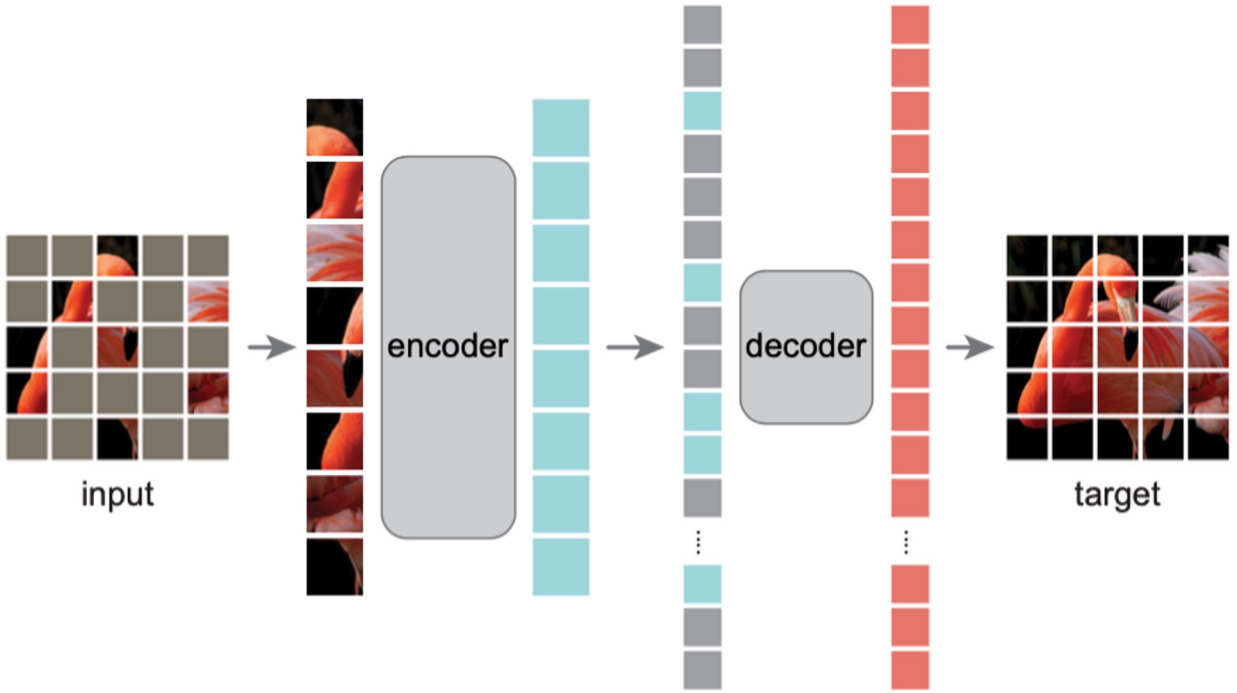
- Input: the full set of patches consisting of (i) encoded visible patches, and (ii) mask tokens
- Mask tokens:
  - a shared vector vector that indicates the presence of a missing patch to be predicted
- Add positional embeddings to all tokens in this full set
- The decoder has another series of Transformer blocks.



[Slide credit: Yannis Kalantidis]

MAE decoder

- **The decoder id only used during pre-training**
  Only the encoder is used to produce image representations in the target tasks
- The decoder architecture can be designed in a way that is **independent of the encoder design**.
  - Typically, the decoder is narrower and shallower than the encoder.

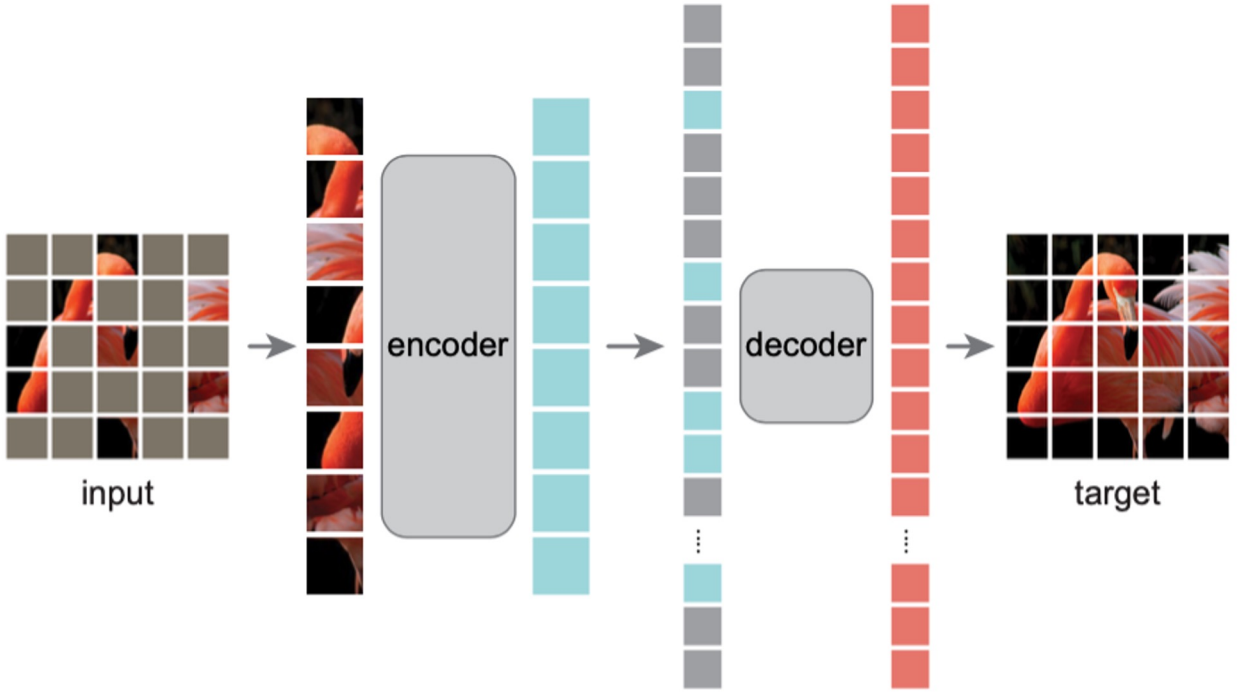


[Slide credit: Yannis Kalantidis]

Reconstruction target

Predict the **pixel values** for each masked patch

- Output is a vector of pixel values representing a patch.
- Loss function computes the **mean squared error** (MSE) between the reconstructed (predicted) images and the original images, in the pixel space.
- The loss is only computed on the masked patches, similar to BERT



[Slide credit: Yannis Kalantidis]

Evaluation de la généralisation d’une représentation

Apprentissage continu de représentations visuelles

2023-2024

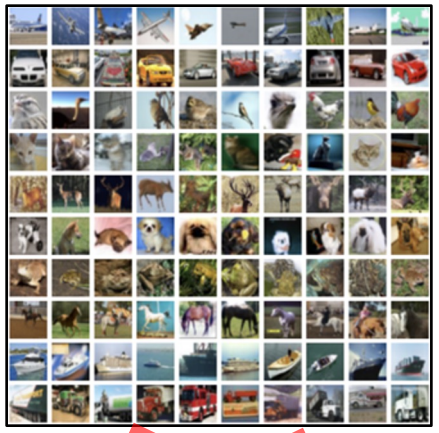
Learning transferable visual representation

Proxy task

Self-Supervised Classification

Images

IMAGENET



labels

Model

Visual representations

Target tasks

Image Classification

Object Detection

Instance Segmentation

Image Retrieval

Evaluation of visual representations

Self-Supervised Classification

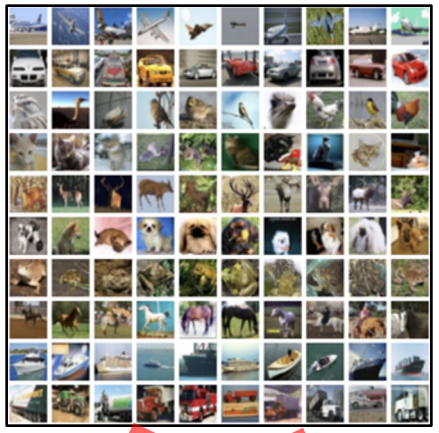
Images

Model

Visual representations

Proxy task

IMAGENET



labels

??

How should we evaluate learnt visual representations, when we don't know the task yet?

Evaluation of visual representations

Self-Supervised Classification

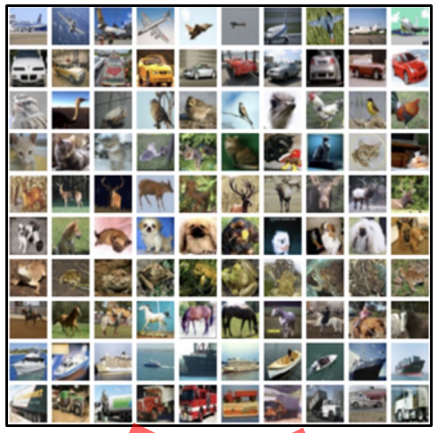
Images

Model

Visual representations

Proxy task

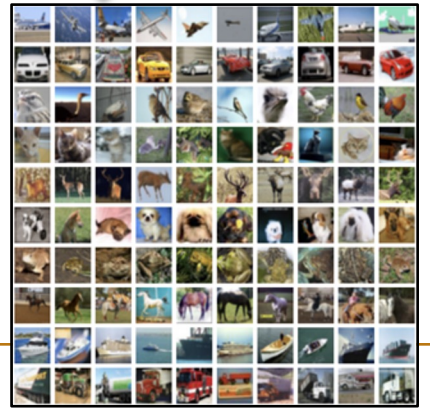
IMAGENET



labels

Target tasks

Image Classification



+ labels

Standard evaluation: **on ImageNet itself** !

Two main evaluation modes:

- Linear probing: train logistic regression on top
- Train an MLP on top

How well does all this generalize beyond ImageNet?

Evaluation of the generalization capability of visual representations

Self-Supervised Classification

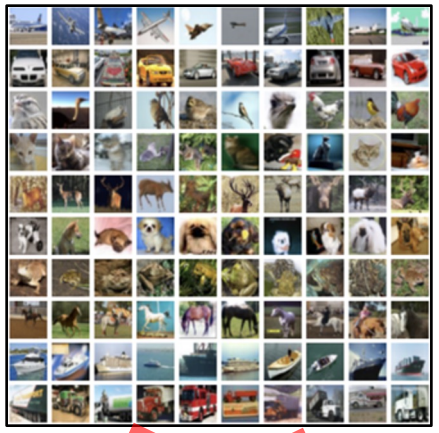
Images

Model

Visual representations

Proxy task

IMAGENET



labels

Most papers:

Measure performance on (many) other datasets

Not very principled ..

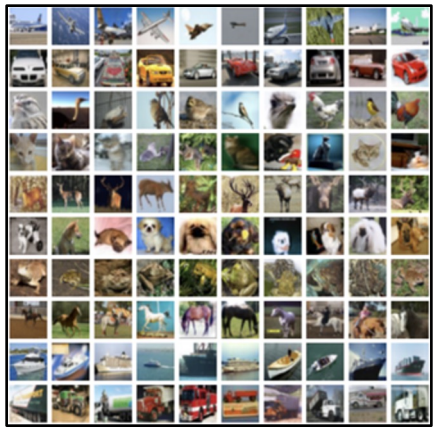
Evaluation of the generalization capability of visual representations

Self-Supervised Classification

Images

Proxy task

IMAGENET



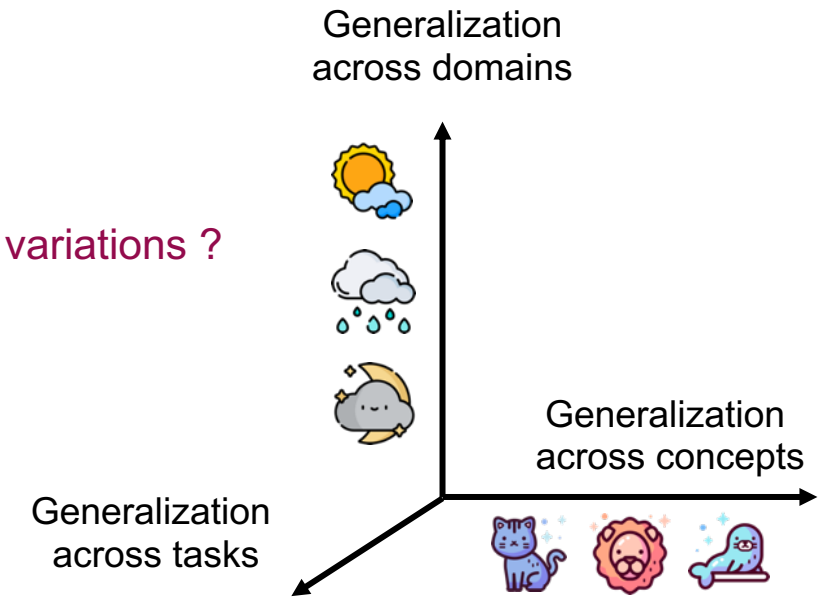
Model

Visual representations

Measure performance

on “each dimension / set of properties that can vary”

Dimensions possibles des variations



Q: Entre la tâche prétexte et la tâche cible, quelles variations peut-on observer ?

Q: Les descripteurs / le modèle que nous venons d'apprendre est-il robuste à ces variations ?

- La distribution des images d'entrée peut varier
  → on parle de généralisation à un nouveau domaine
  (lien avec le cours d'adaptation de domaine)

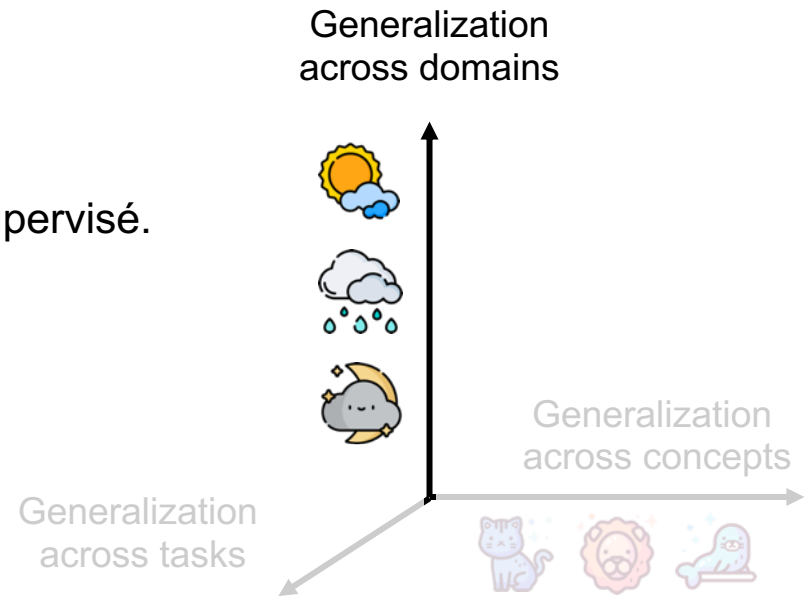
- La nature des scènes et/ou des objects obervés peut changer
  → on parle de généralisation à des classes/concepts non-observés

- La tâche peut changer.
  Les réseaux appris sont utilisables directement pour une tâche de classification, par simple combinaison avec un classifieur (e.g. ajout d'une couche complètement connectée). Soit on apprend seulement cette couche, soit on apprend conjointement cette couche tout en raffinant (*fine-tuning*) le réseau pré-entrainé.
  On peut aussi utiliser ces réseaux pour d'autres tâches, par exemple la détection d'objets, la segmentation sémantique, la prédiction de la profondeur, etc.
  → on parle de généralisation à de nouvelles tâches

Ces trois dimensions de la généralisation sont complètementaires.

Généralisation à un nouveau domaine

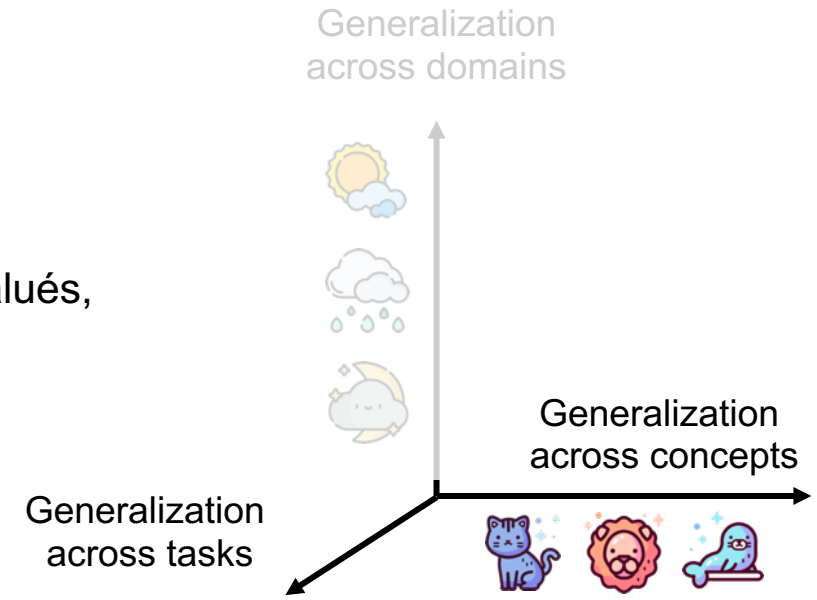
Quelques bases d'évaluation spécifiques existent,
mais cela reste une dimension relativement peu étudiée en apprentissage auto-supervisé.



Exemples de bases d'évaluation possibles:

- DomainNet [Peng@ICCV23]
- How robust is unsupervised representation learning to distribution shift? [Shi@ICLR23]

Généralisation à de nouvelles tâches et à de nouveaux concepts



C’est le type d’évaluation le plus courant.

La plupart des modèles entrainés sur ImageNet (sans ses labels) sont ensuite évalués, après un ré-entrainement minimal :

- en detection d’objet
- en segmentation d’instances d’objets ou en segmentation sémantique
- en recheche d’images

Ils sont également entrainés (souvent seulement par combinaison avec un classifieur linéaire) sur d’autres tâches de classification de natures différentes (classification d’espèces d’oiseaux, de races de chien, de textures, etc.)

Problème

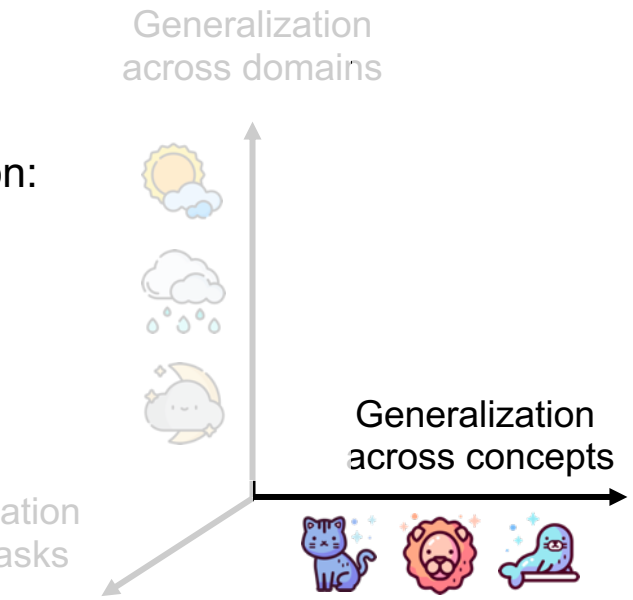
Dans la plupart des cas, certaines classes ont été vues lors de l’apprentissage et d’autres non, et ces statistiques ne sont pas claires: deux des dimensions de la généralisation mentionnées précemment sont mélangées.
Il est donc difficile de tirer des conclusions sur les résultats obtenus.

De plus, la distribution des images elle-même peut également changer, ce qui rajoute une variation sur la troisième dimension (généralisation à un nouveau domaine).

Généralisation à de nouveaux concepts

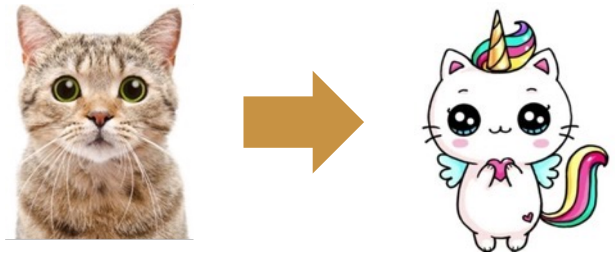
Voyons comment créer une base d'évaluation spécialisée qui se limite à une seule dimension:

la généralisation à de nouveaux concepts.



Narrowing down the evaluation to **Concept Generalization**

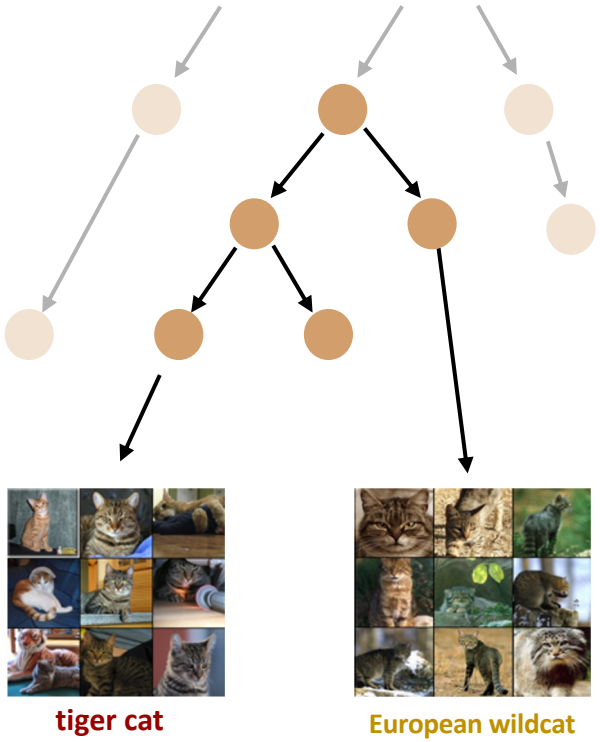
*When training a model on a set of **seen** concepts,*
*how well does it generalize to **new, unseen** set of concepts ?*



Does learning about cats
help us learn faster about caticorns?

Measure the semantic distance between concepts

WordNet Graph



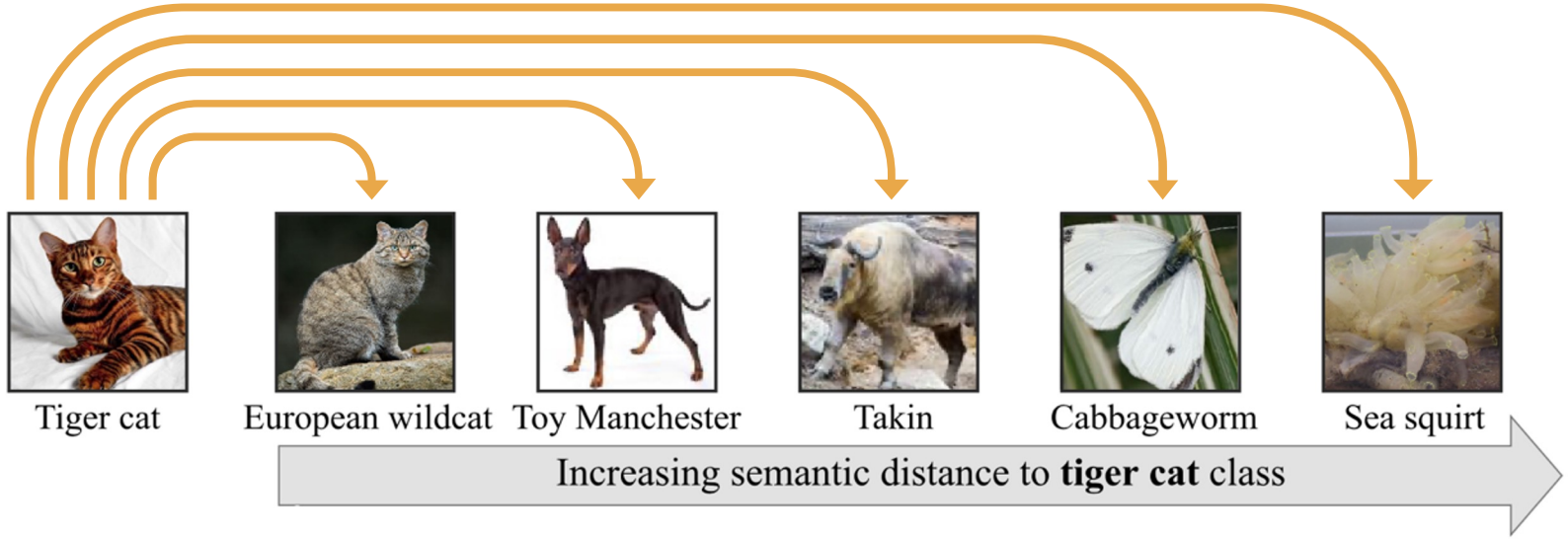
Lin similarity

$$\text{sim}_{\text{Lin}}(c_1, c_2) = \frac{2 \times \text{IC}(\text{LCS}(c_1, c_2))}{\text{IC}(c_1) + \text{IC}(c_2)}$$

Lin similarity in the WordNet Graph

[**Lin**: Lin@ICML1998]

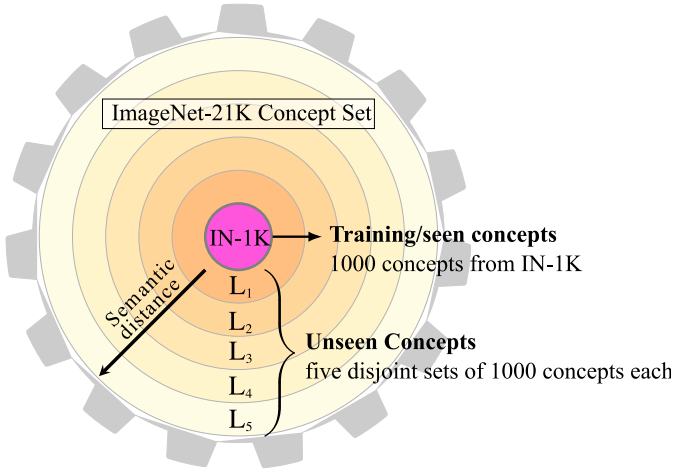
Measure the semantic distance between concepts



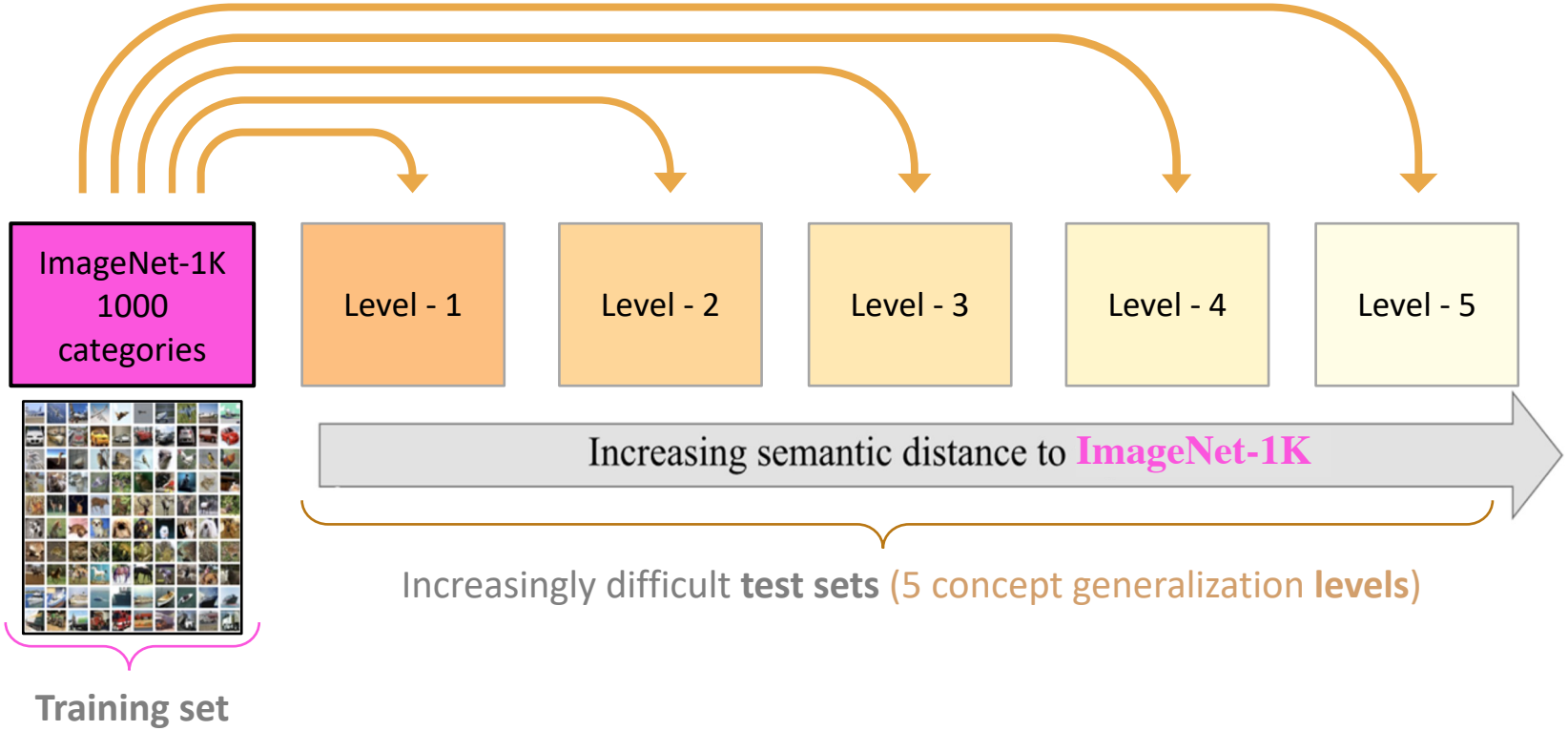
[**Lin**: Lin@ICML1998]

Measure the semantic distance between sets of concepts

[ImageNet: Deng@CVPR2009]



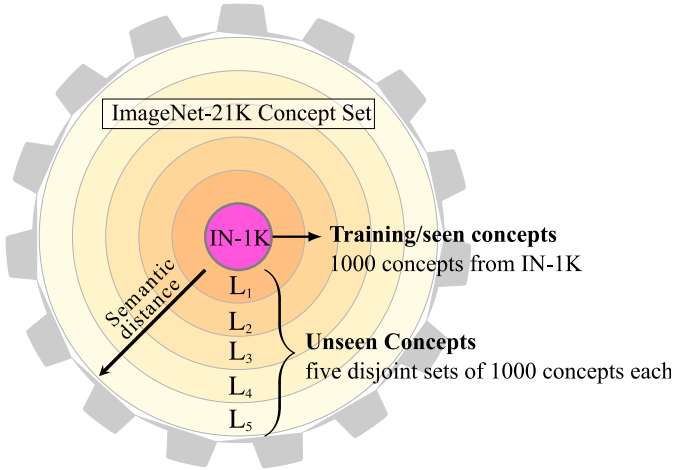
Proposed **CoG** benchmark



Observation

- Recent **self-supervised** approaches generalize better than supervised ones.
- **DINO** has great generalization properties

Concept generalization in visual representation learning
B. Sariyildiz, Y. Kalantidis, D. Larlus, K. Alahari
ICCV 2021



Proposed **CoG** benchmark

ResNet50	**Baseline model** from the torchvision package (25.5M)
Architecture: *Models with different backbone*	
a-T2T-ViT-t-14	Visual transformer (21.5M)
a-DeiT-S	Visual transformer (22M)
a-DeiT-S-distilled	Distilled *a*-DeiT-S (22M)
a-Inception-v3	CNN with inception modules (27.2M)
a-NAT-M4	Neural architecture search model (7.6M)
a-EfficientNet-B1	Neural architecture search model (7.8M)
a-DeiT-B-distilled	Bigger version of *a*-DeiT-S-distilled (87.6M)
a-ResNet152	Bigger version of ResNet50 (60.2M)
a-VGG19	Simple CNN architecture (143.5M)
Self-supervision: *ResNet50 models trained in this framework*	
s-SimCLR-v2	Online instance discrimination (ID)
s-MoCo-v2	ID with momentum encoder and memory bank
s-SwAV	Online clustering
s-BYOL	Negative-free ID with momentum encoder
s-MoCHi	ID with negative pair mining
s-InfoMin	ID with careful positive pair selection
s-OBoW	Online bag-of-visual-words prediction
s-CompReSS	Distilled from SimCLR-v1 (with ResNet50x4)
Regularization: *ResNet50 models with additional regularization*	
r-MixUp	Label-associated data augmentation
r-Manifold-MixUp	Label-associated data augmentation
r-CutMix	Label-associated data augmentation
r-ReLabel	Trained on a "multi-label" version of IN-1K
r-Adv-Robust	Adversarially robust model
r-MEAL-v2	Distilled ResNet50
Use of web data: *ResNet50 models using additional data*	
d-MoPro	Trained on WebVision-V1 ($\sim 2\times$)
d-Semi-Sup	Pretrained on YFCC-100M ($\sim 100\times$), then fine-tuned on IN-1K
d-Semi-Weakly-Sup	Pretrained on IG-1B ($\sim 1000\times$), then fine-tuned on IN-1K
d-CLIP	Trained on WebImageText ($\sim 400\times$)