Contrastive and Masked Representation Learning

Representation Learning

Vladimir Zaigrajew

2025-03-19

Vladimir Zaigrajew - vladimir.zaigrajew.dokt@pw.edu.pl

Tymoteusz Kwieciński - tymoteuszkwiecinski@gmail.com

You can find us in Room 316, MINI, PW

Remember every information you can find on our Github Repo:

Figure 1: QR code to course Github Repo

Figure 2: QR code to our Github Repo

Recap from last lecture

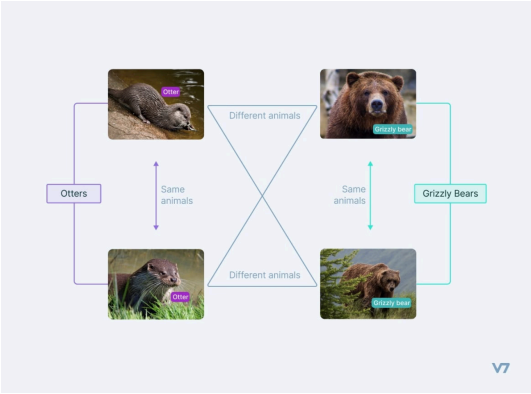
- Self-supervised learning (SSL) is a subfield of representation learning, where model training is done without human-annotated labels.
- SSL process consists of four stages:
  - Unlabeled data collection (X) - Just scrape the web
  - Pretext task definition - Define a task that can be solved using the unlabeled data, for example, next word prediction.
  - Model training - Train the model using the defined pretext tasks. So best on the pretext task create labels for the data and train the model on it.
  - Evaluation - Evaluate the model's performance on specific downstream tasks or transfer learning.
- Before 2020 the techniques were not very effective, but after 2020 the results started to be very good.
- The best SSL models started to outperform the best supervised models.
- No single self-supervised method dominates overall, because SSL models are downstream *task-dependent*, *architecture-dependent*.
- Almost all current best performing models used models that were pretrained with SSL.

Contrastive Learning mimics the way humans learn.

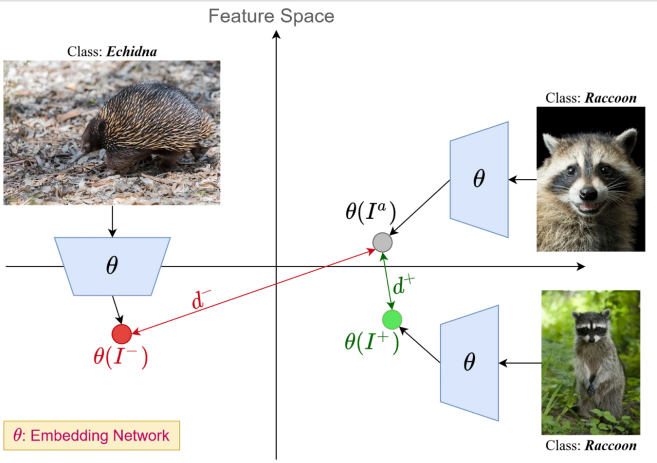
Humans *can* learn just by looking at one image to understand it. But we often **prefer** to learn by comparing images.

We understand concepts better when we see what they are and what they aren't.

For example: When we see a cat, we can understand what a cat is by comparing it to a dog. We can see the differences and similarities between the two animals.



https://www.v7labs.com/blog/contrastive-learning-guide

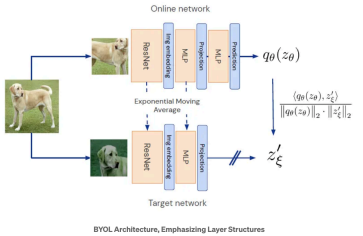


As shown in the example above, two images belonging to the same class lie close to each other in the embedding space ("d+"), and those belonging to different classes lie at a greater distance from each other ("d-"). Thus, a contrastive learning model (denotes by "theta" in the example above) tries to minimize the distance "d+" and maximize the distance "d-."

https://www.v7labs.com/blog/contrastive-learning-guide

Positive Pairs Only

Use the same image but with different augmentations and train the model to learn the same representation for both images.

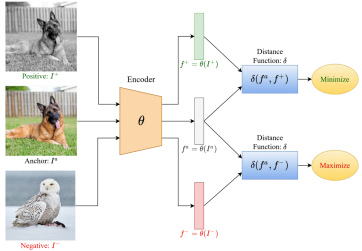


BYOL Architecture, Emphasizing Layer Structures

Grill, Jean-Bastien, et al. "Bootstrap your own latent-a new approach to self-supervised learning." Advances in neural information processing systems 33 (2020): 21271-21284.

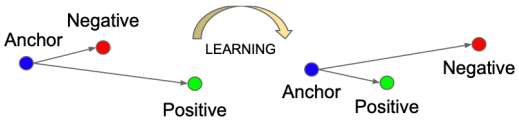
Instance Discrimination

We modify the input data to create two different views of the same image. This allows the model to learn invariant features by contrasting positive pairs against negative pairs (other images).



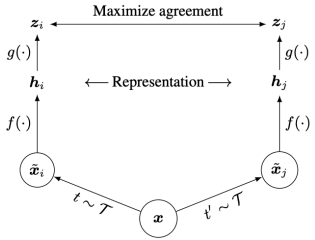
https://www.v7labs.com/blog/contrastive-learning-guide

Triplet loss (2015)



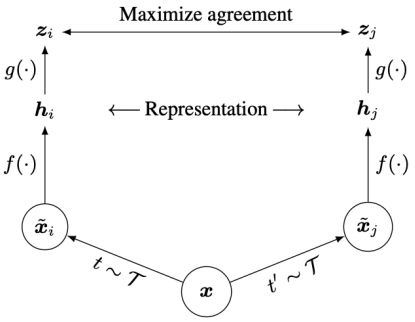
Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

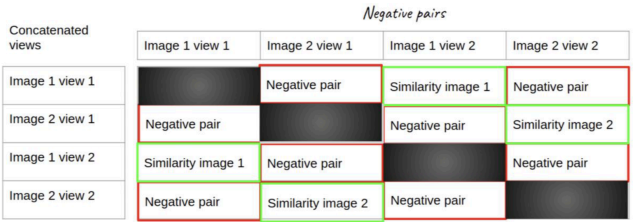
Normalized Temperature-scaled Cross Entropy (NT-Xent) Loss (2020)



Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PmLR, 2020.

Instance Discrimination - SimCLR (2020) nad SimCLRv2 (2020)





How many classes do we have?
How many positive pairs do we have?
How many negative pairs do we have?

Normalized Temperature-scaled Cross Entropy (NT-Xent) Loss

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)},$$

L2 normalization and Temperature scaling

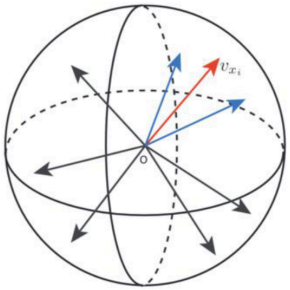
Restricting the output space to the unit hypersphere (unit length)

The softmax distribution → arbitrarily sharp

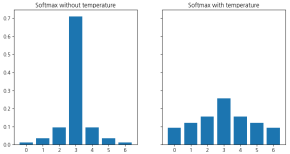
Ignore magnitude, focus on direction (angle) between vectors.

The temperature parameter τ controls the sharpness of the distribution. low values create high-contrast representations where differences are amplified, while high values create more nuanced, graduated similarity relationships.

$\tau = 0.07$ this value is empirically chosen. This value indicate that we want to penalize the model even for slightly dissimilar items (same cats)



128D Unit Sphere



Problems in SSL and Contrastive Learning

Mode collapse: There are scenarios where the model may fail to capture the diversity of the data, collapsing to nearly identical representations Solutions: Stronger augmentations, better negative sampling, regularization techniques

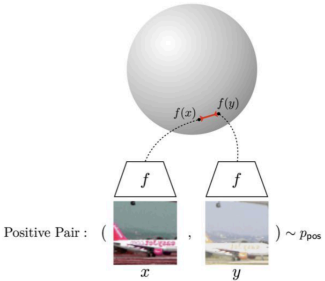
Representation Collapse: The model may learn to ignore the input data and produce similar representations for all inputs. Particularly problematic in methods without negative samples.

Negative Sampling Issues: The choice of negative samples can significantly impact the performance of contrastive learning methods. Poorly chosen negatives can lead to suboptimal representations. Additionally, the number of negatives can be computationally expensive as we need many negative samples for effective learning.

The Cold-Start Problem: Initial representations are poor, making similarity judgments unreliable. Solutions: Larger Batch Sizes, Curriculum Learning, Temperature Scheduling, Leveraging domain knowledge to bootstrap the training.

Properties of contrastive learning

Uniformity: Preserve maximal information.



Alignment: Similar samples have similar features.

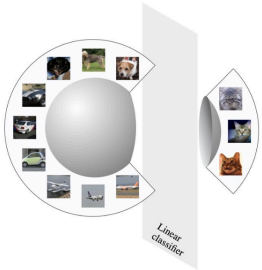
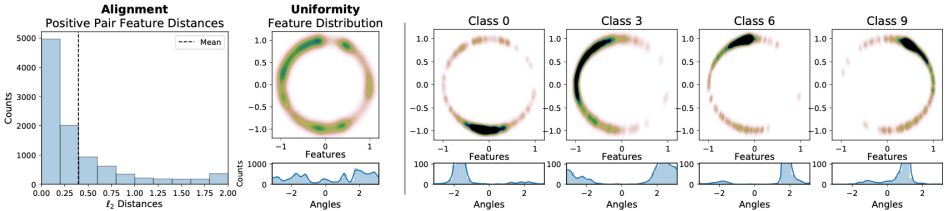
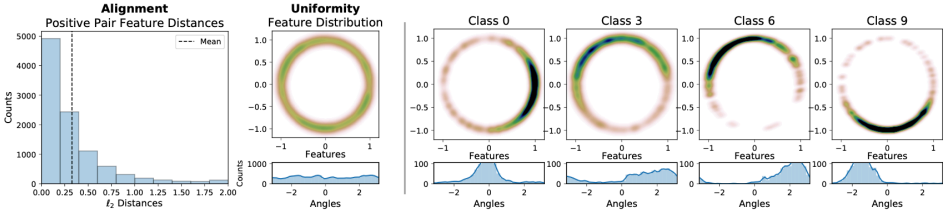


Figure 2: **Hypersphere:** When classes are well-clustered (forming spherical caps), they are linearly separable. The same does not hold for Euclidean spaces.

[0]Wang, Tongzhou, and Phillip Isola. "Understanding contrastive representation learning through alignment and uniformity on the hypersphere." International conference on machine learning. PMLR, 2020.



(b) **Supervised Predictive Learning.** Linear classification validation accuracy: 57.19%.



(c) **Unsupervised Contrastive Learning.** Linear classification validation accuracy: 28.60%.

[0]Wang, Tongzhou, and Phillip Isola. "Understanding contrastive representation learning through alignment and uniformity on the hypersphere." International conference on machine learning. PMLR, 2020.

The same loss as in SimCLR

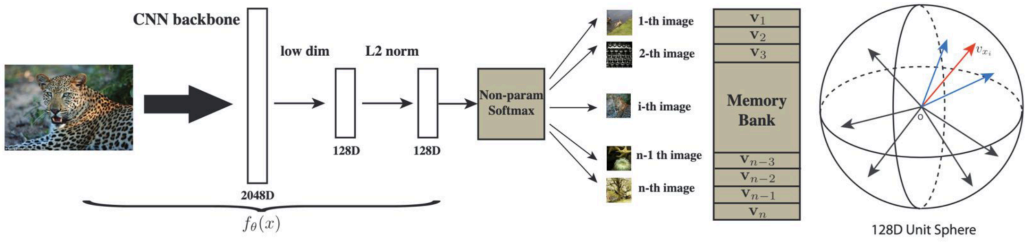


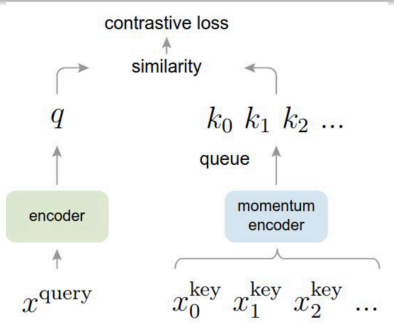
Figure 2: The pipeline of our unsupervised feature learning approach. We use a backbone CNN to encode each image as a feature vector, which is projected to a 128-dimensional space and L2 normalized. The optimal feature embedding is learned via instance-level discrimination, which tries to maximally scatter the features of training samples over the 128-dimensional unit sphere.

[0]Wu, Zhirong, et al. "Unsupervised feature learning via non-parametric instance discrimination." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

Added momentum encoder to the memory bank

> "We hypothesize that such failure is caused by the rapidly changing encoder that reduces the key representations' consistency. We propose an EMA to address this issue. Shortly, encoder embbedings changes faster than the memory bank embeddings."

v2 stronger augmentations and MLP projection head borrowed from SimCLR



$$\mathcal{L}_{q,k^+,\{k^-\}} = -\log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{k^-} \exp(q \cdot k^-/\tau)}.$$

He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020).

MoCO pseudo code

Queue length 65K
Momentum 0.999
T=0.07

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```


He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020).

Redesigned for Vision Transformers

Symmetric contrastive loss

Extra projection head

"We abandon the memory queue, when the batch is sufficiently large" (4096)

Algorithm 1 MoCo v3: PyTorch-like Pseudocode

```
# f_q: encoder: backbone + proj mlp + pred mlp
# f_k: momentum encoder: backbone + proj mlp
# m: momentum coefficient
# tau: temperature

for x in loader: # load a minibatch x with N samples
   x1, x2 = aug(x), aug(x) # augmentation
   q1, q2 = f_q(x1), f_q(x2) # queries: [N, C] each
   k1, k2 = f_k(x1), f_k(x2) # keys: [N, C] each

   loss = ctr(q1, k2) + ctr(q2, k1) # symmetrized
   loss.backward()

   update(f_q) # optimizer update: f_q
   f_k = m*f_k + (1-m)*f_q # momentum update: f_k

# contrastive loss
def ctr(q, k):
   logits = mm(q, k.t()) # [N, N] pairs
   labels = range(N) # positives are in diagonal
   loss = CrossEntropyLoss(logits/tau, labels)
   return 2 * tau * loss
```

Notes: `mm` is matrix multiplication. `k.t()` is `k`'s transpose. The prediction head is excluded from `f_k` (and thus the momentum update).

Chen, Xinlei, Saining Xie, and Kaiming He. "An empirical study of training self-supervised vision transformers." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

Don’t redesign the model, just try to correct the flaws in this approach.

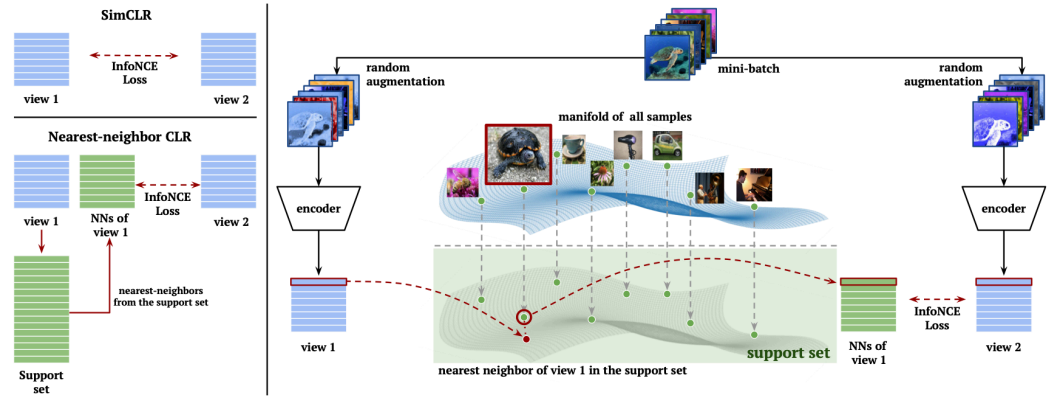


Figure 2: **Overview of NNCLR Training**

[0]Dwibedi, Debidatta, et al. ”With a little help from my friends: Nearest-neighbor contrastive learning of visual representations.” Proceedings of the IEEE/CVF international conference on computer vision. 2021.

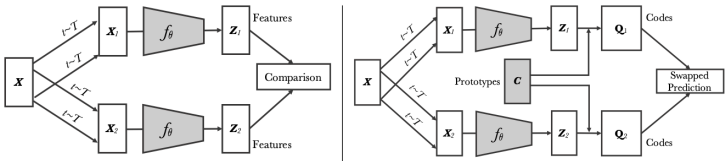
SwAV - Clustering on Positive Pairs

The most representative method among the clustering-based architectures.

Uses the prototypes (C) to cache the center features for clusters.

Codes (Q) represent the extracted feature z by the similarities to each cluster (Solution for mode collapse with Sinkhorn-Knopp algorithm).

Considers only positive pairs in loss functions

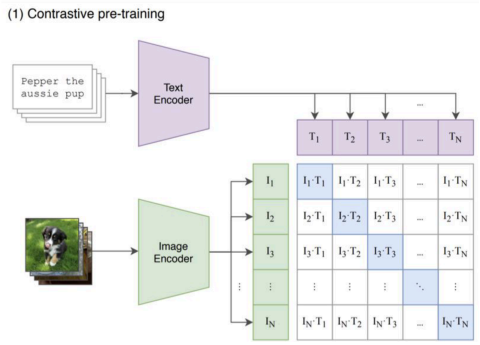


[0]Caron, Mathilde, et al. "Unsupervised learning of visual features by contrasting cluster assignments." Advances in neural information processing systems 33 (2020): 9912-9924.

CLIP (ICML, 2021) - the most used contrastive learning model

Contrastive learning is **not** limited to visual inputs.

The most cited contrastive learning method: **30688**.




```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PmLR, 2021.

The Results of Contrastive Learning

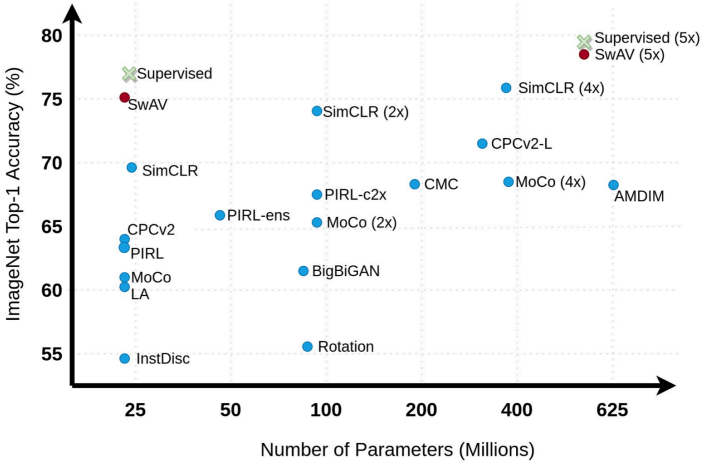


Figure 3: Top-1 classification accuracy of different contrastive learning methods against baseline supervised method on ImageNet

[0]Jaiswal, Ashish, et al. "A survey on contrastive self-supervised learning." Technologies 9.1 (2020): 2.

Masking SSL Theory

Core insight: Learning robust representations by predicting what's missing

Masking imposes structured information bottleneck

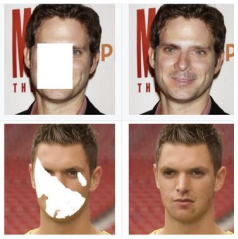
- Forces model to infer relationships between visible/hidden regions
- Prevents trivial solutions and shortcut learning

No negative samples required (unlike contrastive methods)

Human cognition routinely completes patterns from partial information

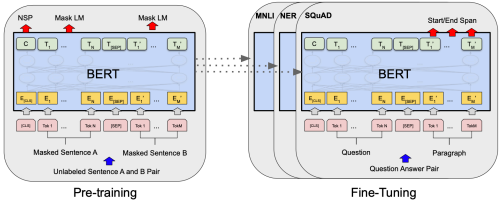
"The dog chased the [MASK] up a tree."
"The dog chased the cat up a tree."

"Students attend [MASK] to gain knowledge and understanding."
"Students attend lectures to gain knowledge and understanding."



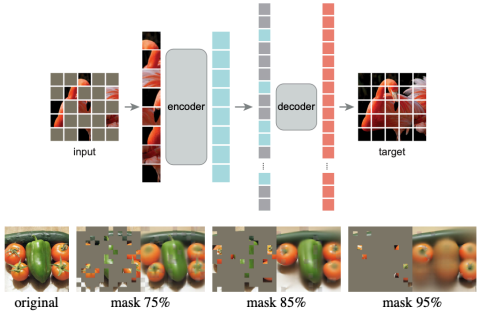
Masked Models for text (2019) and images (2021)

Text (Bert or GPT)



Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). 2019.

Image (MAE)



He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

BERT (2019) - Masked Language Model (MLM)

First major application of masking in language models: **126150 citations**

Uses bidirectional context.

Randomly mask 15% of input tokens:

- 80% replace with [MASK]
- 10% replace with random token
- 10% keep original token
- Prevents model from just memorizing masks

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Tasks	Dev Set MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Table 5: Ablation over the pre-training tasks using the $\text{BERT}_{\text{BASE}}$ architecture. "No NSP" is trained without the next sentence prediction task. "LTR & No NSP" is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT. "+ BiLSTM" adds a randomly initialized BiLSTM on top of the "LTR + No NSP" model during fine-tuning.

[0]Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). 2019.

BERT (2019) - Legacy

- RoBERTa (Liu et al., 2019) - citations: 18311
  - Removed Next Sentence Prediction task
- ALBERT (Lan et al., 2019) - citations: 8456
  - Parameter sharing across layers (85% fewer parameters)
  - Sentence-Order Prediction replacing Next Sentence Prediction
- ELECTRA (Clark et al., 2020) - citations: 4690
  - Add discriminator to predict replaced tokens (GAN-style)
- SpanBERT (Joshi et al., 2020) - citations: 2339
  - Masks contiguous spans rather than random tokens
- BART (Lewis et al., 2020) - citations: 12191
  - Encoder-Decoder Transformer (Combines BERT and GPT)
  - Encoder Processes corrupted text and decoder predict original tokens

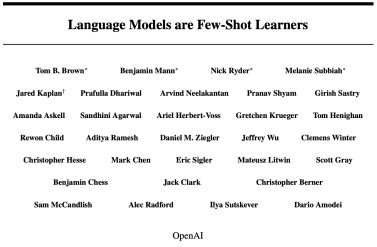
GPT3 (2020) - Generative Pre-trained Transformer 3

The last SSL trained open source GPT model: **41905 citations**

Decoder-only transformer (no encoder component)

Autoregressive language modeling (predicts next token)

175 billion parameters (10x larger than previous models)



Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

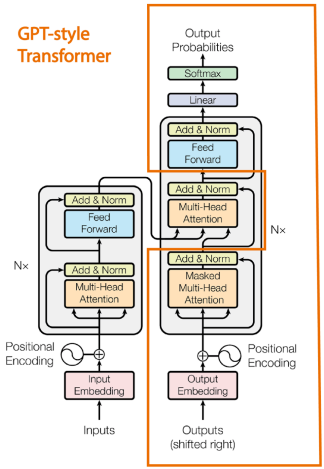
Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

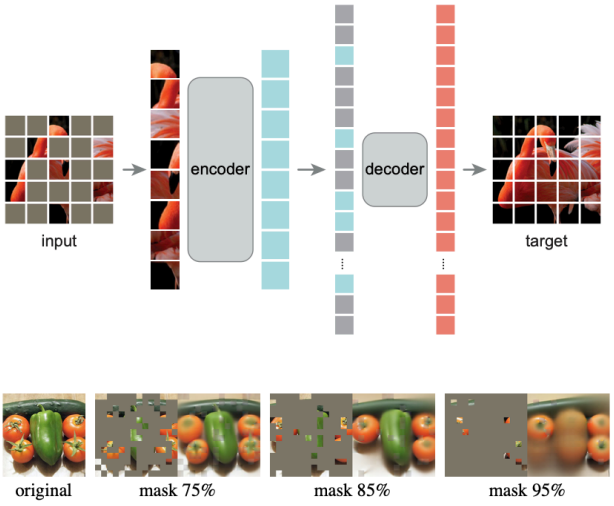
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI



Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.

MAE (CVPR, 2022) - Masked Autoencoder with 9024 citations



case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

(e) **Data augmentation**. Our MAE works with minimal or no augmentation.

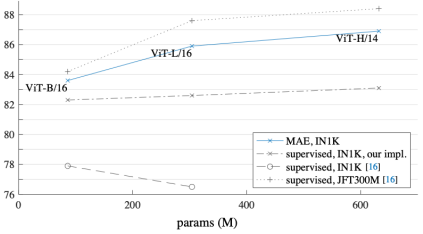


Figure 8. **MAE pre-training *vs*. supervised pre-training**, evaluated by fine-tuning in ImageNet-1K (224 size). We compare with the original ViT results [16] trained in IN1K or JFT300M.

[0]He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

MAE - What is the optimal masking ratio?

Intuition: Vision-based signals are more redundant than natural language (~15% masking ratio)

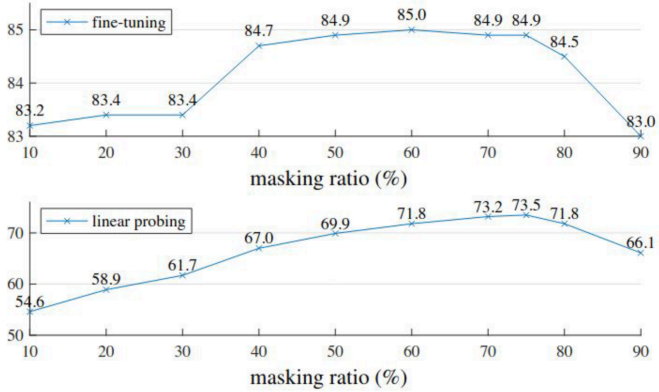


Figure 5. **Masking ratio**. A high masking ratio (75%) works well for both fine-tuning (top) and linear probing (bottom). The y-axes are ImageNet-1K validation accuracy (%) in all plots in this paper.

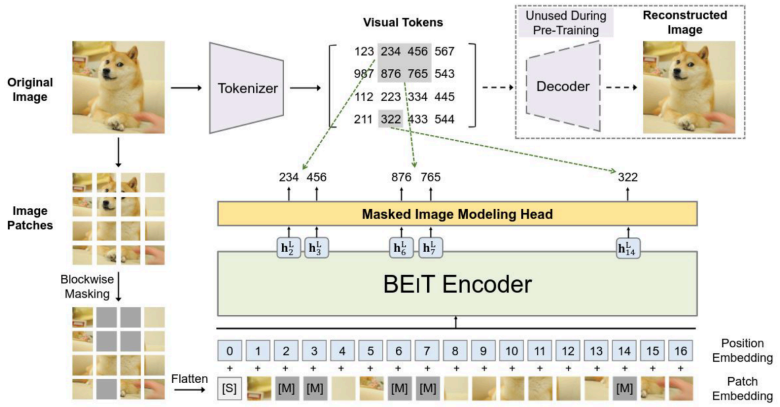
[0]He, Kaiming, et al. "Masked autoencoders are scalable vision learners." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

BEiT (ICLR, 2022) - BERT-style pre-training in vision

Randomly mask some patches and replace them with token[M].

"tokenize" the image to discrete visual tokens latent codes of dVAE (Discrete Variational Autoencoder)

Pretext task: predicting the visual tokens of the original image based on the corrupted image.



Bao, Hangbo, et al. "Beit: Bert pre-training of image transformers." arXiv preprint arXiv:2106.08254 (2021).

- BEit v2 (ICLR,2023):
  - Replaces dVAE with knowledge distillation from teacher model
  - Uses ”soft” visual tokens instead of discrete ones
- BEiT v3 (CVPR,2023):
  - Multiway transformer architecture for both images and text
  - Joint vision-language masking objectives

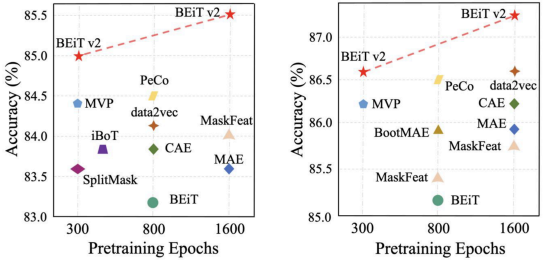


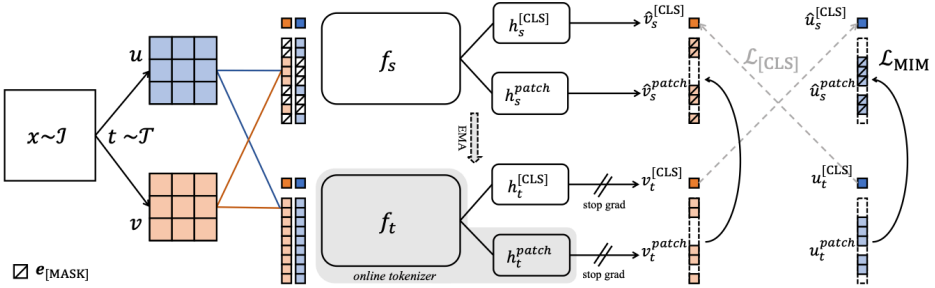
Figure 1: Top-1 fine-tuning accuracy on ImageNet (224 size). **Left**: ViT-B/16. **right**: ViT-L/16.

Image-level: CLS token obtained from different views of the same image.

$$\mathcal{L}_{[\mathrm{CLS}]} = -P_{\boldsymbol{\theta}'}^{[\mathrm{CLS}]}(\boldsymbol{v})^{\mathrm{T}} \log P_{\boldsymbol{\theta}}^{[\mathrm{CLS}]}(\boldsymbol{u})$$

MIM: CE(masked_student, teacher_unmasked) of the **same view**

$$\mathcal{L}_{\mathrm{MIM}} = -\sum_{i=1}^{N} m_i \cdot P_{\boldsymbol{\theta}'}^{\mathrm{patch}}(\boldsymbol{u}_i)^{\mathrm{T}} \log P_{\boldsymbol{\theta}}^{\mathrm{patch}}(\hat{\boldsymbol{u}}_i).$$



[0]Zhou, Jinghao, et al. "ibot: Image bert pre-training with online tokenizer." arXiv preprint arXiv:2111.07832 (2021).

BYOL (NeurIPS, 2020) - Cherry on the cake

No negative pairs, No large batch size dependency, No memory bank

Computational efficiency during training

Asymmetric architecture with predictor

Strong empirical results

Requires careful architectural design to avoid collapse (batch normalization)

Sensitive to hyperparameters

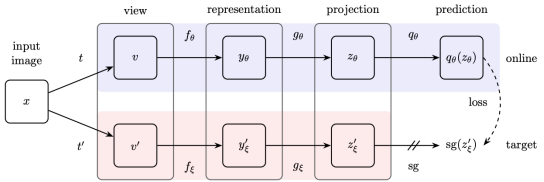


Figure 2: BYOL's architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where θ are the trained weights, ξ are an exponential moving average of θ and sg means stop-gradient. At the end of training, everything but f_θ is discarded, and y_θ is used as the image representation.

Grill, Jean-Bastien, et al. "Bootstrap your own latent-a new approach to self-supervised learning." Advances in neural information processing systems 33 (2020): 21271-21284.