# Data2Vec Paper Review

Othman Mohamed Othman
2002395
Mahmoud Aboud Mohammad Nada
2002387

Ain Shams University
Faculty of Engineering
Computer and System Department

# Introduction

Data2Vec is a paper published by Meta AI in 2022. It proposes a general framework for self-supervised learning which generalizes and uses the same learning algorithm for Speech, NLP, and Computer Vision. The main idea is to predict latent representations of the input data based on a masked version of the input using a standard Transformer architecture. This means that the final output for each sample of the input will be a latent representation carrying information about the sample as well as its context. These representations will be used eventually in target down-stream tasks. Instead of predicting modality-specific targets such as words, visual tokens, or units of human speech which are local in nature, data2vec predicts contextualized latent representations that contain information from the entire input.

## Motivation

### Self Supervised

Traditional Machine Learning Solutions depend on the presence of labeled data, from which the model can learn the target mapping by learning from data-label pairs. As AI models evolve and increase in size, they require more and more data to perform more complex tasks. Labeling and annotation of data is a costly process that requires a lot of resources. Recently, efforts have been directed more and more toward Self Supervised Learning (SSL). Instead of using supervised training on labeled data, SSL correlation and dependency between unlabeled data. The neural network learns in two steps. First, the task is solved based on pseudo-labels which help to initialize the network weights. This means that the network can learn insightful information about the input from the data structure alone. Then, down-stream tasks are learned through supervised or unsupervised learning.

### UniModal

While the general idea of self-supervised learning is identical across modalities, the actual algorithms and objectives differ widely because they were developed with a single modality in mind. Recently, there have been many Algorithms that have become popular for self-supervised tasks such as BERT for NLP tasks, Wav2Vec and HuBERT for Speech Recognition, Self Supervised Vision Transformers (ViT) for Computer Vision. Researchers have always aimed to reach Artificial General Intelligence (AGI) which means the ability of AI models to learn and understand intellectual tasks in the same way humans can do. Researchers have found out that human infants learn speech, audio, and vision in the same way. Data2Vec follows the same approach and proposes a uni-modal algorithm for training the transformer for the 3 modes of data. The same architecture and training procedure are shown to be valid for the 3 fields of data, with only modification of the preprocessing layer or the feature extraction techniques.

### Contextualized:

The idea of using latent representation in Deep Learning has been popular for a long time. Many classical approaches aim to produce latent representation for the input samples which carries information about the data structure. It has been shown that these latent representations will be way more powerful if they embed the context as well and not only the discrete samples of the data. Some approaches have been directed toward latent representations embedding sub context of the data. Data2vec aims to achieve full contextualization where for each sample, the model has access to the full input and can base the attention on the full sequence of the data.

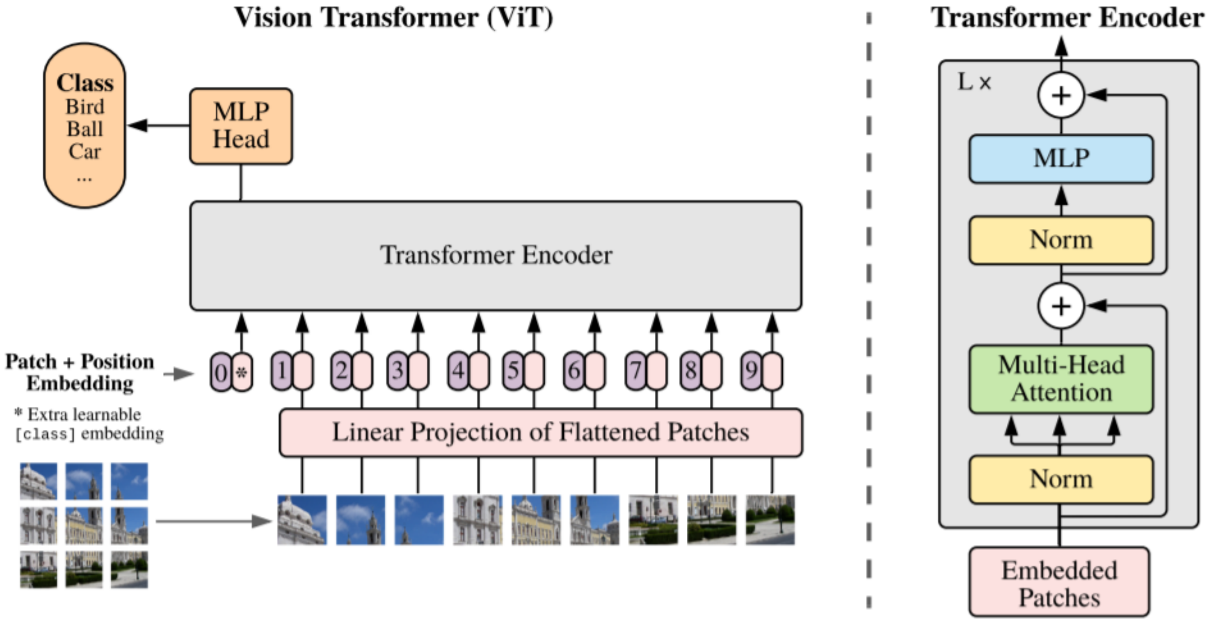
# Related Work

## Computer Vision

### Vision Transformer (ViT)

The 2020 paper: "An Image is Worth 16x16 Pixels: Transformers for Image Recognition at Scale" by Dosovitskiy, A., et al (Google) introduced the Vision Transformer, which at first just seemed like a cool extension of NLP Transformers but which has now proved to be very effective for computer vision tasks.

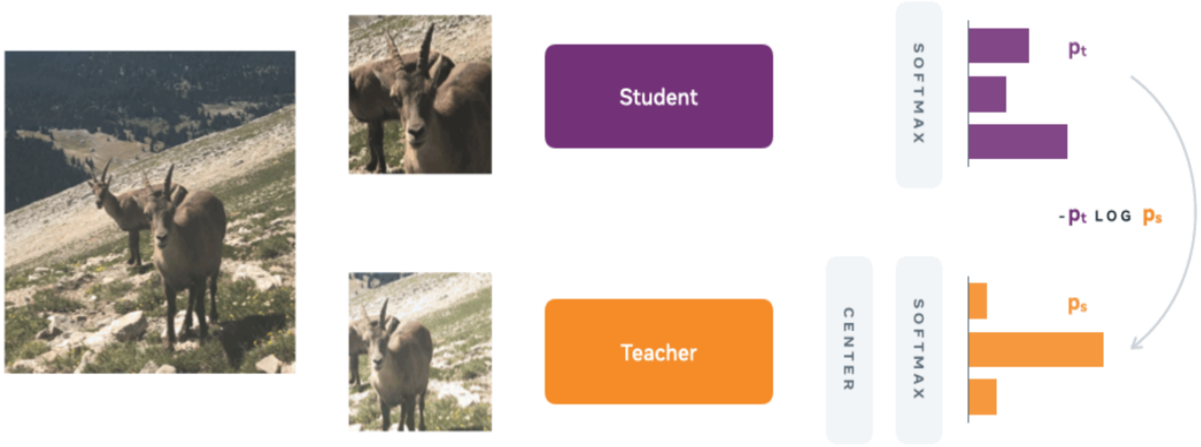
An image is split into patches (16x16x3), which are then flattened and put into a lower-dimensional embedding space. An extra token is added to each vector (i.e. linearly-embedded patch) to denote its relative location in the image (i.e. positional embedding), and another, learnable token is added to the entire sequence of vectors to represent the class. The sequence of vectors is fed to the standard Transformer encoder, which has been modified with an extra fully-connected layer at the end for doing classification.



## DINO

DINO, a new self-supervised system by Facebook AI, can learn incredible representations from unlabeled data. Below is a video visualizing its attention maps and we see the model was able to automatically learn class-specific features leading to accurate unsupervised object segmentation. It was introduced in their paper “Emerging Properties in Self-Supervised Vision Transformers”

A Student ViT learns to predict global features in an image from local patches supervised by the cross-entropy loss from a momentum Teacher ViT’s embeddings while doing centering and sharpening to prevent mode collapse



## Bootstrap Your Own Latent (BYOL)

BYOL uses two same encoder networks referred to as online and target network for obtaining representations and reducing the contrastive loss between the two representations.

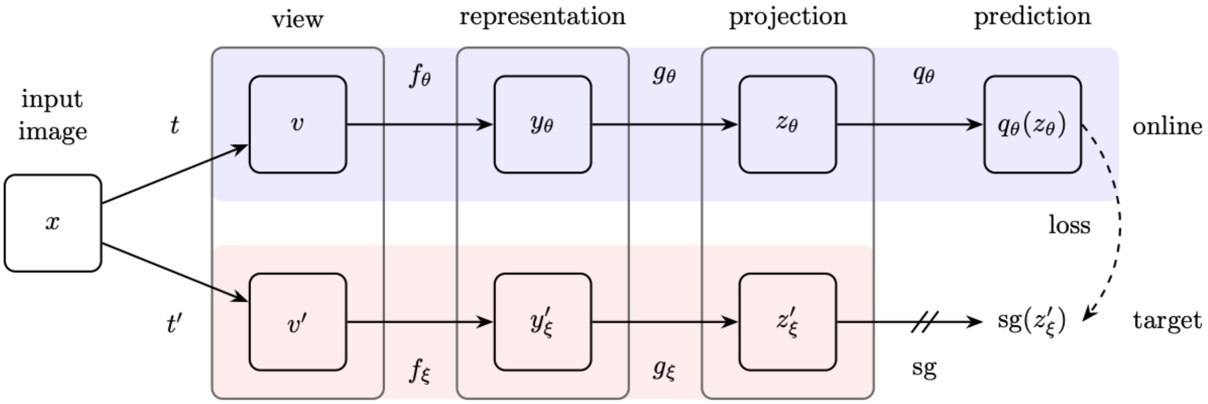


Figure 2: BYOL’s architecture. BYOL minimizes a similarity loss between $q_\theta(z_\theta)$ and $\mathrm{sg}(z'_\xi)$, where $\theta$ are the trained weights, $\xi$ are an exponential moving average of $\theta$ and sg means stop-gradient. At the end of training, everything but $f_\theta$ is discarded,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

BYOL minimizes the distance between representations of each sample and a transformation of that sample. Examples of transformations include: translation, rotation, blurring, color inversion, color jitter, Gaussian noise, etc. (I’m using images as a concrete example here, but BYOL works with other data types, too.) We usually train using several different types of transformations, which can be applied together or independently. In general, if you want your model to be invariant under a particular transformation, then it should be included in your training.

# Natural Language Processing

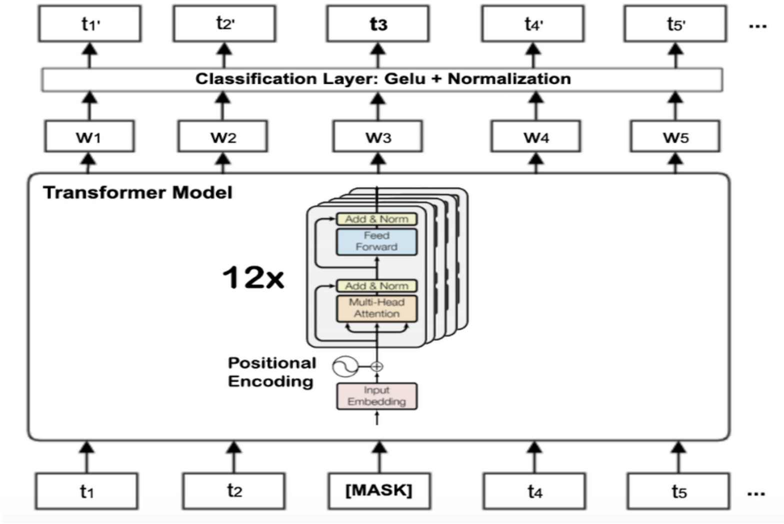
## BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers is an open-source machine learning framework for natural language processing (NLP). BERT is designed to help computers understand the meaning of ambiguous language in a text by using surrounding text to establish context. The BERT framework was pre-trained using text from Wikipedia and also can be fine-tuned with a question and answer datasets

BERT, which stands for Bidirectional Encoder Representations from Transformers, is based on Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based on their connection. (In NLP, this process is called attention.)

The objective of Masked Language Model (MLM) training is to hide a word in a sentence and then have the program predict what word has been hidden (masked) based on the hidden word's context. The objective of Next Sentence Prediction training is to have the program predict whether two given sentences have a logical, sequential connection or whether their relationship is simply random.

The goal of any given NLP technique is to understand human language as it is spoken naturally. In BERT's case, this typically means predicting a word in a blank.

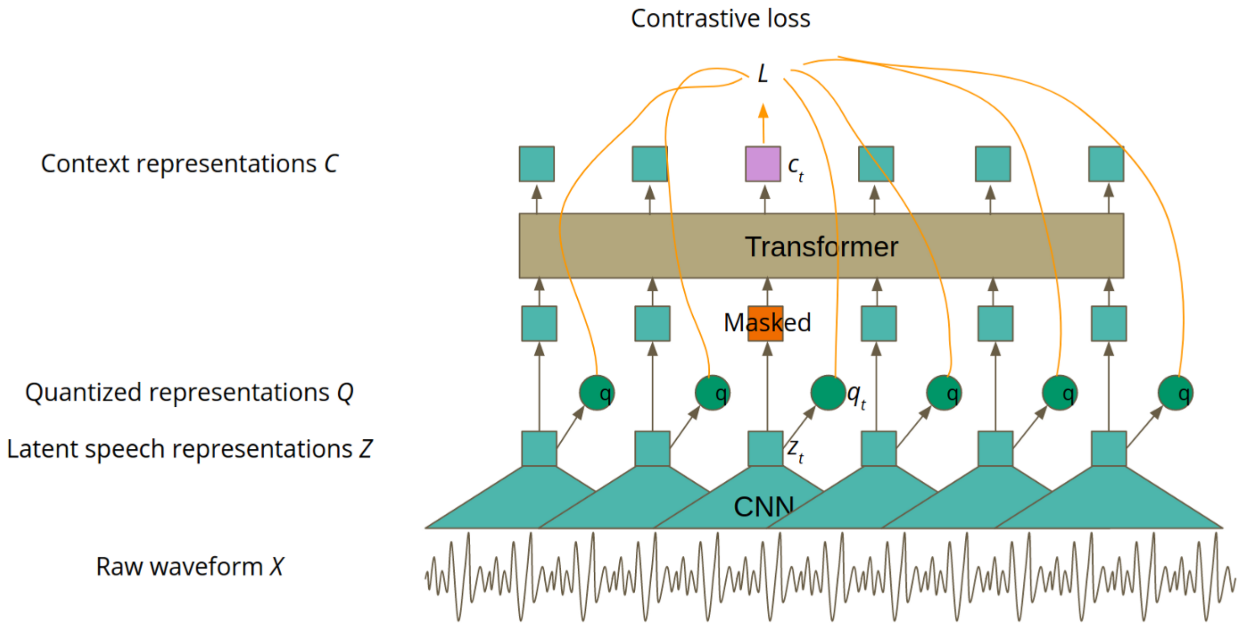


# Speech

## Wav2vec

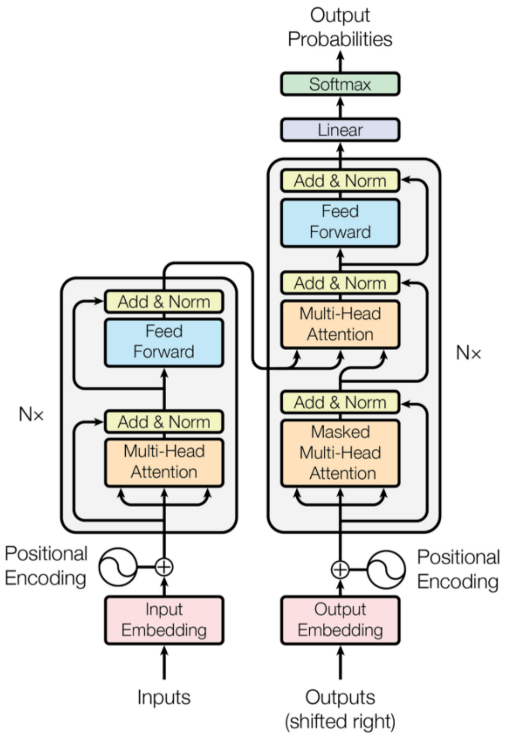
Wav2Vec 2.0 is one of the current state-of-the-art models for Automatic Speech Recognition due to self-supervised training which is quite a new concept in this field. This way of training allows us to pre-train a model on unlabeled data which is always more accessible. Then, the model can be fine-tuned on a particular dataset for a specific purpose. As the previous works show this way of training is very powerful

the model is trained in two phases. The first phase is in a self-supervised mode, which is done using unlabeled data and aims to achieve the best speech representation possible. You can think about that in a similar way as you think of word embeddings. Word embeddings also aim to achieve the best representation of natural language. The main difference is that Wav2Vec 2.0 processes audio instead of text. The second phase of training is supervised fine-tuning, during which labeled data is used to teach the model to predict particular words or phonemes


Contrastive loss
$L$
Context representations $C$
$c_t$
Transformer
Masked
Quantized representations $Q$
q
q
$q_t$
q
q
q
Latent speech representations $Z$
$z_t$
CNN
Raw waveform $X$

# Method

## Model Architecture



The Standard transformer architecture is used. A transformer is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data with modality-specific encoding. Like recurrent neural networks (RNNs), transformers are designed to process sequential input data. However, unlike RNNs, transformers process the entire input all at once. The attention mechanism provides context for any position in the input sequence based on the effect of each part of the sequence on the target.

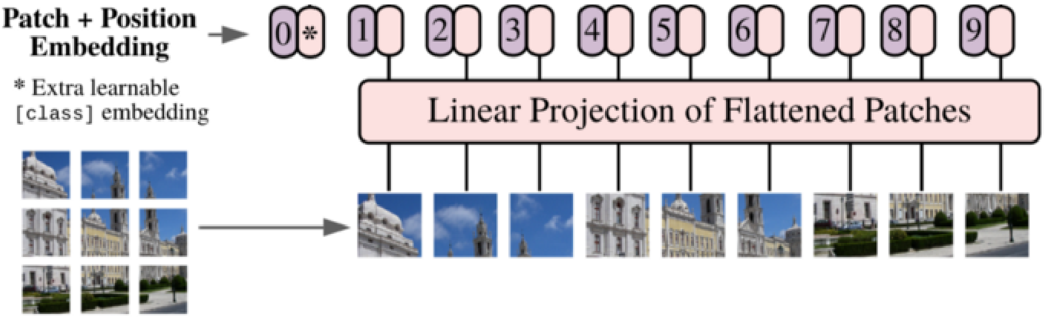
Transformer networks have been very popular to deal with data that has a correlation between different samples like Images, Audio Clips, or Text Documents. Data2vec builds on that and uses the same transformer network across modalities. Although the network is unified for all modalities, Modality specific encoding is needed for each type of data to preprocess inputs before feeding them to the network.

## Preparing Data

For each modality, a specific procedure is carried out to embed the input data into relative tokens to be fed to the network

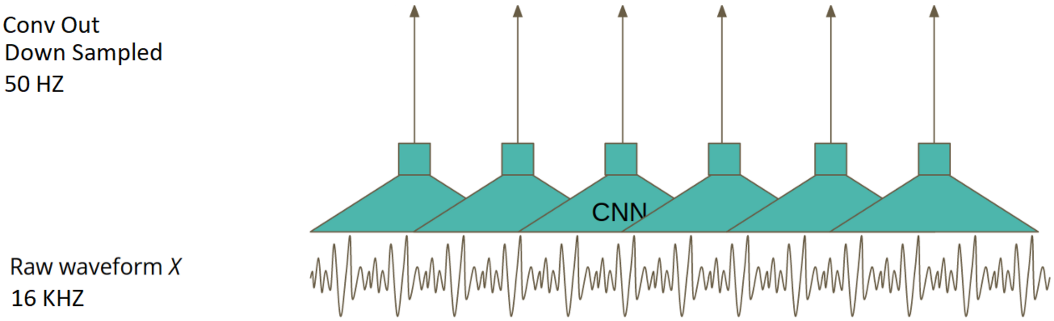
### Computer Vision

Vision Transformer (ViT) encoding strategy is used. This means encoding the image as a sequence of patches, each of size 16x16 pixels. These batches are then input to a linear transformation to produce the final encoding which will be input to the network.


Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

### Speech

For a speech signal, Multi-Layer 1-D Convolutions are used. Each convolution layer down samples the input leading to eventually down sampling 16kHz waveform to 50 Hz samples. This encoding technique is inspired by wav2vec 2.0.



### Text

Text is pre-processed to obtain sub-words (predefined sub units forming words) which are then embedded in distributional space via learned embedding vectors. The same approach is used in RoBERTa, which is a robust implementation of BERT.

## Masking

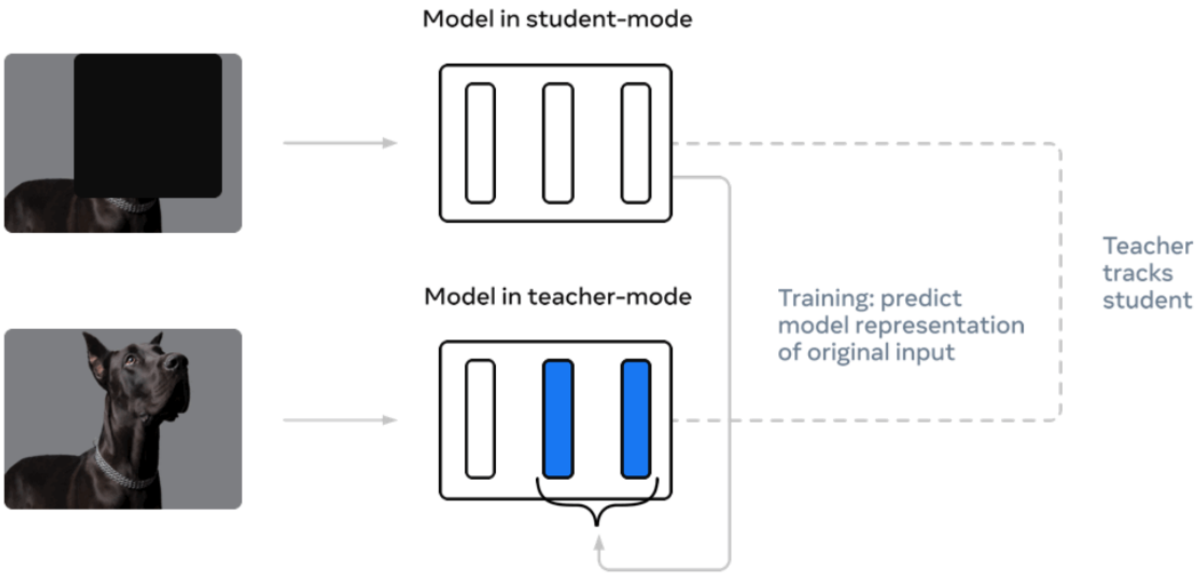
Masking is performed after preparing the data and after the input is embedded as the relevant sequence of tokens. A part of the input units in the token sequence is masked by replacing them

with a learned MASK embedding token before feeding the sequence to the Transformer network. This masking step is crucial for the latent representation training as the student network predicts the representation of the masked tokens based on the learned weights.
For Computer Vision, Block wise masking strategy is followed similar to what was proposed in BERT pre-training for Vision Transformers.
For the speech portion of the speech, representations are masked, and for text, the percentage of the text tokens is masked.

## Training Targets

Data2vec is training models to predict their own representations of the input data, regardless of the modality. By focusing on these representations — the layers of a neural network — instead of predicting visual tokens, words, or sounds, a single algorithm can work with completely different types of input. This removes the dependence on modality-specific targets in the learning task. Directly predicting representations is not straightforward, and it required defining a robust normalization of the features for the task that would be reliable in different modalities.

This method uses a teacher network to first compute target representations from an image, a piece of text, or a speech utterance. Next, we mask part of the input and repeat the process with a student network, which then predicts the latent representations of the teacher. The student model has to predict representations of the full input data even though it has a view of only some of the information. The teacher network is identical to the student model but with weights that are slightly out of date

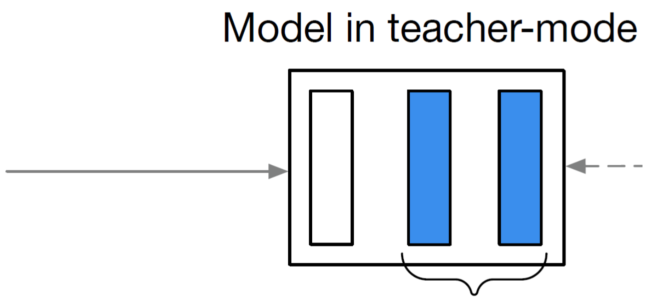


### Teacher parameterization

They use a schedule for T that linearly increases this parameter from $\tau_0$ to the target value $\tau_e$

over the first $\tau_n$ updates after which the value is kept constant for the remainder of the training.

$$\Delta \leftarrow \tau\Delta + (1 - \tau)\,\theta$$



### Targets

- Training targets are constructed based on the output of the top-K blocks of the teacher network for time steps which are masked in student-mode.
- Normalizing the targets helps prevent the model from collapsing into a constant representation for all time steps and it also prevents layers with high norm to dominate the target features. (speech: instance norm, CV&NLP: parameter-less layer norm)
- The output of block $l$ at time-step $t$ is denoted as $a_l^t$. We apply normalization to each block to obtain $a_l^t$ before averaging the top K blocks $y_t = \frac{1}{k}\sum_{l=L-K+1}^{L} a_t^l$
- VICreg also addresses this problem but above strategy perform well with fewer hyperparameters.

## Objective (Loss)

$$\mathcal{L}(y_t, f_t(x)) = \begin{cases} \frac{1}{2}(y_t - f_t(x))^2/\beta & |y_t - f_t(x)| \leq \beta \\ (|y_t - f_t(x)| - \frac{1}{2}\beta) & \text{otherwise} \end{cases}$$

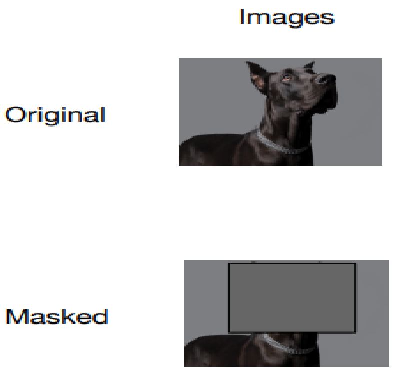
- Smooth L1 loss to regress these targets.
- β controls the transition from a squared loss to an L1 loss.
- The advantage of this loss is that it is less sensitive to outliers, however, we need to tune the setting of β.

# Experimental Results

They experiment with two model sizes: data2vec Base and data2vec Large, containing either L = 12 or L = 24 Transformer blocks with H = 768 or H = 1024 hidden dimension (with 4 x H feed-forward inner-dimension). EMA updates are performed in fp32 for numerical stability.

## Computer Vision

- They embed images of 224x224 pixels as patches of 16x16 pixels like the normal ViT.
- They are also masking blocks of multiple adjacent patches where each block contains at least 16 patches with a random aspect ratio.
- 60% of the image is masked



- They use randomly applied resized image crops, horizontal flipping, and color jittering
- They use the same modified image both in teacher mode and student mode.
- 2 models are trained ViT-B and ViT-L

| **Hyper-parameter** | **ViT-B** | **ViT-L** |
|---|---|---|
| Epochs | 800 | 1600 |
| Batch size | 2048 | 8192 |
| Warm-up learning rate / Epochs | 0.002/40 | 0.001/80 |
| Learning rate | Learning rate is annealed following the cosine schedule | |
| optimizer | Adam (cousin scheduling) | Adam (cousin scheduling) |
| $\beta$ | 2 | 2 |
| $\tau$ | 0.9998 | 0.9998-0.9999 |
| $K$ | 6 | 6 |

- For image classification we mean-pool the output of the last Transformer block and input it to a softmax-normalized classifier

## Speech

Models are implemented in fairseq which is a sequence modeling toolkit written in PyTorch that allows researchers and developers to train custom models for translation, summarization, language modeling, and other text generation tasks.

it takes as input 16 kHz waveform which is processed by a feature encoder (Baevski et al., 2020b) containing seven temporal convolutions with 512 channels, strides (5,2,2,2,2,2,2) and kernel widths (10,3,3,3,3,2,2).

This results in an encoder output frequency of 50 Hz with a stride of about 20ms between each sample, and a receptive field of 400 input samples or 25ms of audio. The raw waveform input to the encoder is normalized to zero mean and unit variance.

Approximately 49% of all time steps are to be masked for a typical training sequence

They optimize with Adam, with a peak learning rate of $5 \times 10^{-4}$ for data2vec Base.

They follow the fine-tuning regime of wav2vec 2.0 whose hyper-parameters depend on the labeled data setup.

## Natural Language Processing

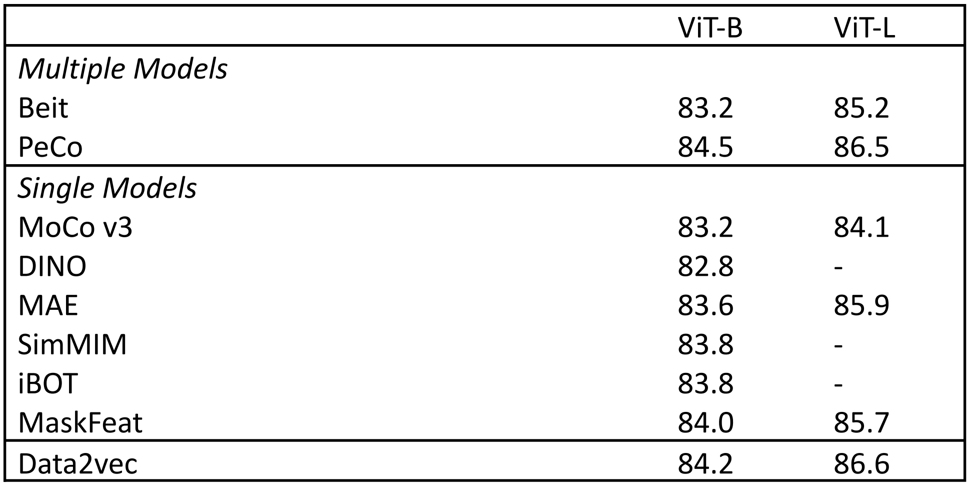
- They build on the BERT re-implementation RoBERTa also available in fairseq.
- The input data is tokenized using a byte-pair encoding of 50K types and the model learns an embedding for each type.
- Once the data is embedded, they apply the BERT masking strategy to 15% of uniformly selected tokens: 80% are replaced by a learned mask token, 10% are left unchanged and 10% are replaced by randomly selected vocabulary tokens.
- They do not use the next-sentence prediction task.
- They also consider the wav2vec 2.0 strategy of masking spans of four tokens.

# Results

Experiments on the major benchmarks of speech recognition, image classification, and natural language understanding show that the proposed framework outperforms or at least is comparable to the previous state-of-the-art in all the domains. Experimental results show data2vec to be effective in all three modalities, setting a new state of the art for ViT-B with single models and ViT-L on ImageNet-1K, improving over the best prior work in speech processing on speech recognition (HuBERT and Wav2Vec) and outperforming a like for like RoBERTa baseline on the GLUE natural language understanding benchmark.

## Computer Vision

Data2Vec was pre-trained on the images from the ImageNet-1K training set, and the resulting model was fine-tuned for image classification using the labeled data from the same data set. The model was evaluated and compared with previous self-supervised models in terms of top-1 accuracy of the classification task on the validation set. In the paper, it is distinguished between the results based on a single self-supervised model, and results that train a separate visual tokenizer on additional data or distill other self-supervised models. It is shown that data2vec outperforms prior work with ViT-B and ViT-L in the single model setting and all prior work for ViT-L, ViT-L stands for Vision Transformer Large and ViT-B stands for Vision Transformer Base.



| | ViT-B | ViT-L |
|---|---|---|
| *Multiple Models* | | |
| Beit | 83.2 | 85.2 |
| PeCo | 84.5 | 86.5 |
| *Single Models* | | |
| MoCo v3 | 83.2 | 84.1 |
| DINO | 82.8 | - |
| MAE | 83.6 | 85.9 |
| SimMIM | 83.8 | - |
| iBOT | 83.8 | - |
| MaskFeat | 84.0 | 85.7 |
| Data2vec | 84.2 | 86.6 |

## Speech Recognition

Data2Vec was pre-trained on total of 960 hours of speech data from LibriSpeech Dataset. LibriSpeech contains relatively clear speech audio. To compare with different sizes of resources, The fine-tuning process for Automatic Speech Recognition was carried out with the different amounts of labeled data ranging from 10 minutes to 960 hours. This is a popular evaluation strategy for self-supervised ASR systems. It is worth noting that for supervised approaches, it is nearly impossible to train a meaningful model with labeled data as little as 10 minutes of audio. Word Error Rate (WER) was the comparison metric. It is shown that data2vec outperforms the previous models for the small sizes of labeled data with the biggest gain being for the 10 minutes labeled setup. When the labeled data gets larger, the gain becomes less and for resource, reach labeled data like 960 hours setting, the performance of data2vec is still comparable.

| | Amount of Labeled Data | | | | |
|---|---|---|---|---|---|
| | 10m | 1h | 10h | 100h | 960h |
| Base Models | | | | | |
| Wav2vec 2.0 | 15.6 | 11.3 | 9.5 | 8.0 | 6.1 |
| HuBERT | 15.3 | 11.3 | 9.4 | 8.1 | - |
| WavLM | - | 10.8 | 9.2 | 7.7 | - |
| Data2vec | 12.3 | 9.1 | 8.1 | 6.8 | 5.5 |
| Large Models | | | | | |
| Wav2vec 2.0 | 10.3 | 7.1 | 5.8 | 4.6 | 3.6 |
| HuBERT | 10.1 | 6.8 | 5.5 | 4.5 | 3.7 |
| WavLM | - | 6.6 | 5.5 | 4.6 | - |
| Data2vec | 8.4 | 6.3 | 5.3 | 4.6 | 3.7 |

## Natural Language Processing

Data2vec is pre-trained with the same training set up as BERT. Pre-Training was carried out on the data from the Books Corpus and English Wikipedia data. General Language Understanding Evaluation (GLUE) benchmark is used to evaluate the model. GLUE includes tasks for natural language inference, sentence similarity, grammatical analysis, and sentiment analysis. Data2vec is finetuned separately on labeled data from each task and the average accuracy from the development set is calculated. It is shown that data2vec outperforms both BERT and its reimplementation RoBERTa for the average GLUE score when the span of 4 tokens is masked with a masking probability of 0.35 (wav2vec style masking).

| | Average GLUE Score |
|---|---|
| BERT | 80.7 |
| RoBERTa | 82.5 |
| Data2vec | 82.7 |
| + wav2vec 2.0 masking | 82.9 |

# Discussion

## Modality-specific feature extractors and masking

Despite the unified learning regime, they still use modality-specific feature extractors and masking strategies. This makes sense given the vastly different nature of the input data.

## Structured and contextualized targets

For NLP, data2vec is the first work that does not rely on predefined target units and that the features of the training targets are contextualized since the features are built with self-attention over the entire unmasked input in teacher mode.

BYOL and DINO also use latent target representations based on the entire input, their focus is on learning transformation-invariant representations instead of structural information within a sample.

## Representation collapse.

A common issue with algorithms that learn their own targets is representation collapse. This occurs when the model produces very similar representations for all masked segments which results in a trivial task

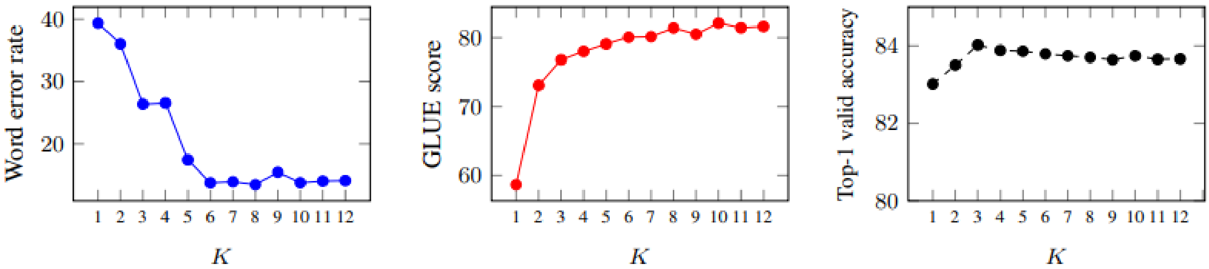
They found that collapse is most likely to happen in the following scenarios:

- First, the learning rate is too large or the learning rate warmup is too short which can often be solved by tuning the respective hyperparameters.
- Second, $\tau$ is too low which leads to student model collapse and is then propagated to the teacher.
- Third, we found collapse to be more likely for modalities where adjacent targets are very correlated and where longer spans need to be masked, e.g., speech. We address this by promoting variance through normalizing target representations over the sequence or batch.

# Ablation

## K averaging effect

Data2vec framework is built on the transformer architecture. For most of the previous work that produces latent representation from transformers, only the output of the last block is used. However, experiments on wav2vec 2.0 have shown that the top layer of the network doesn't perform as well as the middle layers for downstream tasks.
Experiments on-base data2vec (12 layers) were carried out to check the effect of averaging the output of K layers instead of using the output of the top layer only. It was shown that for all the 3 domains, targets based on multiple layers improve over using only the top layer (K = 1). Using all layers is generally a good choice and only slightly worse than a carefully tuned value of K.

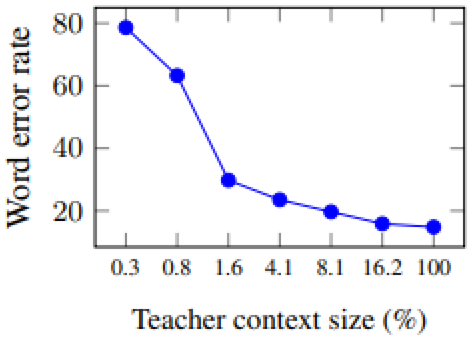


(a) Speech (b) NLP (c) Vision

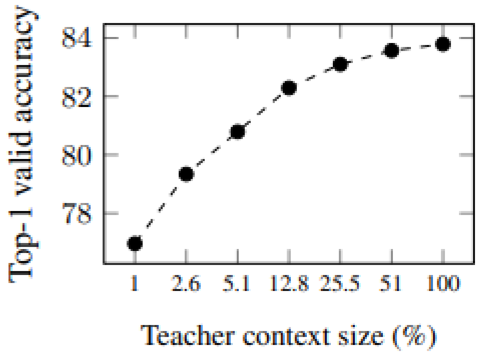
## Target Contextualization

For The Training, Teacher Student Architecture is used. The Teacher network is based on self-attention over all the input sequences. This produces a fully contextualized output which means that for every sample the output carries information about the whole sequence. This is different from most of the previous work where the Teacher has access to only sub part of the context.

It is shown that this access to full context leads to the best results. Experiments have been carried out on both audio and vision data using different Context sizes and the results supported the claim that full access to full context leads to the best results.



(a) Speech

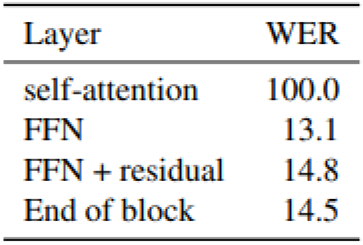


(b) Vision

## Target Feature Type

Transformers blocks contain several layers which can each serve as targets. In the paper, the authors try different features from different layers of the transformer block. It is shown that the output of the feedforward block works the best for the latent representation.

This is shown by fine-tuning on speech recognition task and calculating the Word Error Rate for features extracted from different layers.



| Layer | WER |
|---|---|
| self-attention | 100.0 |
| FFN | 13.1 |
| FFN + residual | 14.8 |
| End of block | 14.5 |

# Conclusion

Self Supervised Approaches have greatly improved the ability of Artificially Intelligent models to learn about the real world without the need for much-labeled data. Self Supervised Learning has been widely used to produce powerful contextualized latent representations of data that can be used in downstream tasks. Data2vec Shows that a single self-supervised learning regime can be effective for vision, speech, and language. A single learning method for multiple modalities will make it easier to learn across modalities and future work may investigate tasks such as audio-visual speech recognition or cross-modal retrieval.

# Strength and Limitations

## Strengths

- Unified Architecture and Training Algorithm for different modalities.
- Strong Contextualized Latent Representation, Achieving State of the Art in various downstream tasks.
- Self Supervised Training, which means they need for labeled data is less than supervised approaches which decrease resources cost.

## Limitations

- Modality Specific encoding is needed for each model.
- The models are based on transformer architecture which means model sizes are large and require strong computation power to train.

# Training Vs Inference

- data2vec follows the teacher-student procedure for training. That means that there are 2 networks involved in the training process which are the teacher producing target representations, and the student producing predicted representation from the masked sequence. This leads to the need for large memory specs while training to be able to load the 2 networks. However, at inference time or test time, the teacher network is not needed. The student network will be trained well and it will be used alone to produce the latent representations.

- During Training, the Percentage of the input samples input to the student network is masked. This is an essential step for the student to be able to learn the required embedding of the input. During Inference, This masking procedure is dropped. The full input sequence is fed to the student network.