Efficient Self-supervised Learning with Contextualized Target Representations for Vision, Speech and Language

Alexei Baevski [1] **Arun Babu** [2] **Wei-Ning Hsu** [2] **Michael Auli** [2]

Abstract

Current self-supervised learning algorithms are often modality-specific and require large amounts of computational resources. To address these issues, we increase the training efficiency of data2vec, a learning objective that generalizes across several modalities. We do not encode masked tokens, use a fast convolutional decoder and amortize the effort to build teacher representations. data2vec 2.0 benefits from the rich contextualized target representations introduced in data2vec which enable a fast self-supervised learner. Experiments on ImageNet-1K image classification show that data2vec 2.0 matches the accuracy of Masked Autoencoders in 16.4x lower pre-training time, on Librispeech speech recognition it performs as well as wav2vec 2.0 in 10.6x less time, and on GLUE natural language understanding it matches a retrained RoBERTa model in half the time. Trading some speed for accuracy results in ImageNet-1K top-1 accuracy of 86.8% with a ViT-L model trained for 150 epochs. Models and code are available at `www.github.com/pytorch/fairseq/tree/master/examples/data2vec`.

1. Introduction

Self-supervised learning has been an active research topic which resulted in much progress across several areas such as computer vision (Grill et al., 2020; Bao et al., 2021; He et al., 2021), natural language processing (NLP; Radford et al. 2018; Devlin et al. 2019; Raffel et al. 2019; Brown et al. 2020), and speech processing (van den Oord et al., 2018; Schneider et al., 2019; Baevski et al., 2020b; Hsu et al., 2021; Baevski et al., 2021; Babu et al., 2022). However, algorithms are often designed with a single modality in mind which makes it unclear whether the same learning mechanisms generalize across modalities. To this end, recent work has introduced unified model architectures (Jaegle et al., 2021b;a) and training objectives which function identically in different modalities (Baevski et al., 2022).

Self-supervised models have benefited from increased scale in model capacity and training datasets (Brown et al., 2020) as well as large amounts of computational training effort (Hoffmann et al., 2022) which resulted in interesting emerging properties (Wei et al., 2022). And while the resulting models are excellent few-shot learners (Brown et al., 2020), the preceding self-supervised learning stage is far from efficient: for some modalities, models with hundreds of billions of parameters are trained which often pushes the boundaries of what is computationally feasible.

In this paper, we present data2vec 2.0 which improves the compute efficiency of self-supervised learning with contextualized target prediction (Baevski et al., 2022) by using an efficient data encoding (He et al., 2021), a fast convolutional decoder and by reusing target representations for multiple masked versions of each sample (Assran et al., 2022; Girdhar et al., 2022). The algorithm uses the same learning objective for each modality but trains separate models for each one using the Transformer architecture with different feature encoders depending on the input modality. We follow Baevski et al. (2022) by creating latent contextualized representations with a teacher model based on unmasked training examples which are regressed by a student model whose input is a masked version of the sample (Figure 1)

Target contextualization enables capturing information about the entire sample, e.g., for text, these targets can represent the different meanings of a word depending on the context. This is more difficult for conventional non-contextualized targets which use a single set of features to represent the different meanings of a word. At first glance, the creation of contextualized targets with a separate teacher appear to be an additional step that slows model training but our efficiency improvements suggest that contextualized targets result in a richer learning task and faster learning.

Experiments demonstrate efficiency improvements of between 2-16x at similar accuracy on image classification,

[1]Character AI, work done while at Meta AI [2]Meta AI. Correspondence to: Alexei Baevski <alexei.b@gmail.com >, Michael Auli <michaelauli@meta.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

speech recognition and natural language understanding.

2. Related Work

Self-supervised learning for NLP, Speech and Vision. There has been much work on self-supervised learning for individual modalities such as NLP where models segment text into sub-word units and define the learning task based on these units by predicting either the next token for causal models (Radford et al., 2018; Brown et al., 2020; Chowdhery et al., 2022) or by predicting masked tokens for bi-directional models (Devlin et al., 2019; Baevski et al., 2019).

For speech processing, models either reconstruct the audio signal (Eloff et al., 2019; Liu et al., 2021) or solve a learning task based on discretizing short and overlapping windows of the speech signal, either in a left-to-right fashion (van den Oord et al., 2018; Schneider et al., 2019; Baevski et al., 2020a) or using masked prediction (Baevski et al., 2020b; Hsu et al., 2021; Chen et al., 2021a).

In computer vision, there has been a shift towards Vision Transformer architectures (ViT; Dosovitskiy et al. 2020) and masked prediction methods that can be very efficient by not encoding masked patches (He et al., 2021). There are also approaches that learn based on discrete visual tokens (Bao et al., 2021; Peng et al., 2022). Other approaches are based on self-distillation (Grill et al., 2020; Caron et al., 2021) and online clustering (Caron et al., 2020b).

Related work to our multi-mask training regime (§3.3) includes Caron et al. (2020a) which creates multiple crops from the same image which contrasts to our approach of creating multiple masked versions of the same training example. Jing et al. (2022) also experiment with applying different masks in the context of convolutional neural networks to a training example but find that data augmentations such as cropping and flipping outperform this masking strategy. Finally, Wu et al. (2022b) also consider multiple masks but they predict representations of the entire training sample and do not average multiple layers.

Generalized Architectures and Learning Objectives. Another trend is the unification of neural network architectures that can process data from different modalities using the same network (Jaegle et al., 2021b;a). This is complemented by work on unifying the self-supervised learning objective for vision, speech, and text in data2vec (Baevski et al., 2022). A distinguishing characteristic of data2vec is that it is trained by predicting contextualized target representations which contain features from the entire input example compared to the limited information of a particular time-step or patch.

Joint Multi-modal Learning. While data2vec and the current work are trained for each modality individually, there has been considerable work on training joint modality models which can represent multiple modalities within the same model. This includes models trained on images and text (Radford et al., 2021; Singh et al., 2021; Wang et al., 2021; Alayrac et al., 2022), speech and text (Shi et al., 2022), or video/audio/text (Akbari et al., 2021).

Efficient Self-supervised Learning. After the success of BERT in NLP, follow on work include a more lightweight training objective to increase efficiency (Clark et al., 2020) and work on reducing the model capacity through weight sharing (Lan et al., 2019) which resulted in faster training speed. In computer vision, He et al. (2021) introduced the idea of not processing masked patches in the encoder network which increased training speed and Assran et al. (2022) used this idea in a joint embedding architecture to achieve label efficient self-supervised learning. There is also work on sparse attention to increase efficiency (Li et al., 2021). For speech, more efficient feature encoder models and time-step squeezing has helped to improve efficiency (Wu et al., 2022a; Vyas et al., 2022).

3. Method

Our approach builds on data2vec (Baevski et al., 2022) and we first describe the major shared techniques including predicting contextualized target representations (§3.1). Similar to Masked Autoencoders (MAE; He et al. 2021), we encode only non-masked portions of a sample and use a decoder model to predict target representations for the masked portions but instead of using a Transformer-based decoder, we use a smaller convolutional decoder which we find to be easier and faster to train (§3.2). To amortize the computational overhead of creating contextualized target representations, we reuse each target for multiple masked versions of a training sample (§3.3) and instead of random masking or block masking, our inverse block masking strategy ensures that contiguous regions of the sample are preserved to provide more context for student predictions (§3.4).

3.1. Contextualized Target Prediction

Instead of reconstructing local windows of the the raw input data (He et al., 2021), or predicting discrete representations thereof (Bao et al., 2021), we predict representations of the teacher network incorporating information from the entire input sample. This leads to a richer training task where targets are specific to a particular training sample. Contextualized targets are built via the self-attention mechanism of a Transformer-based teacher model which encodes the unmasked training sample (Paulus et al., 2017; Vaswani et al., 2017) and the training targets are a weighted sum of

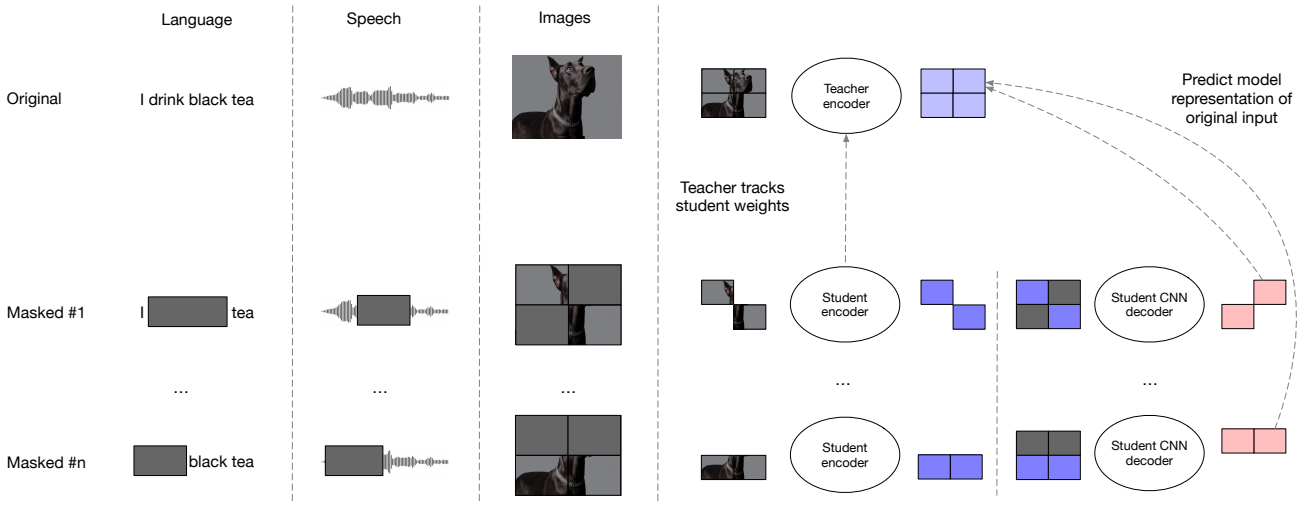


Figure 1. data2vec 2.0 uses the same learning objective for different modalities (but trains different models for each). We first create a *contextualized target* representation based on the unmasked training sample using the teacher model whose weights are an exponentially moving average of the student model. Target representations are contextualized due to self-attention in Transformer models. The same target representation is predicted by the student model for different masked versions of the training example, thereby amortizing the computational cost of creating target representations. Masked portions of the training sample are not encoded (He et al., 2021).

all features in the sample.

Target Representations and Learning Objective. Training targets are based on averaging the top K FFN blocks of the teacher. Before averaging, activations are normalized using instance normalization (Ulyanov et al., 2016).[1] The training task is for the student network to regress these targets based on the masked version of the sample.

Teacher Weights. The teacher weights Δ are an exponentially moving average of the student encoder weights θ (Grill et al., 2020): $\Delta \leftarrow \tau\Delta + (1-\tau)\,\theta$ where τ follows a linearly increasing schedule from a starting value τ_0 to a final value τ_e over τ_n updates, after which the value is kept constant (Baevski et al., 2022).

Learning Objective. We use an L2 loss based on the target representation from the teacher network $\mathbf{y}$ and the student network prediction $\mathbf{f}(\mathbf{x})$. This is a simplification compared to the Smooth L1 loss used in Baevski et al. (2022) and we found it to work well across modalities.

3.2. Model Architecture

Similar to data2vec (Baevski et al., 2022), our model uses modality-specific feature encoders and a Transformer architecture where the latter makes up the the bulk of the model weights (Vaswani et al., 2017). For computer vision, we use a patch mapping of 16x16 pixels as feature encoder (Dosovitskiy et al., 2020), for speech a multi-layer convolutional network following van den Oord et al. (2018); Baevski et al. (2020b; 2022) and for text we use embeddings learned based on byte-pair encoding (Sennrich et al., 2016).

Asymmetric Encoder/Decoder Architecture. In a first step, we use the teacher network to encode all parts of the unmasked training sample in order to create training targets (§3.1). Next, we mask part of the sample (§3.4) and embed it with the student encoder. To improve efficiency, we encode only unmasked patches or time-steps of a training example which leads to a large speed-up compared to encoding all parts of the sample (He et al., 2021), depending on the amount of masking. The output of the student encoder is then merged with fixed representations for the masked portions and fed to a decoder network. To represent the masked tokens, we found it sufficient to use random Gaussian noise compared to a learned representation (He et al., 2021).[2] The decoder network then reconstructs the contextualized target representation of the teacher network for time-steps which are masked in the student input.

Convolutional Decoder Network. We use a lightweight decoder consisting of D convolutions, each followed by layer normalization (Ba et al., 2016), a GELU activation function (Hendrycks & Gimpel, 2016), and a residual connection (He et al., 2015). For sequential data such as speech

[1] Layer normalization (Ba et al., 2016) of the averaged targets can be useful for some modalities such as speech and vision.

[2] We also experimented with adding positional embeddings but found that they do not improve results.

and text we use 1-D convolutions and for images we use 2-D convolutions, each parameterized by groups to increase efficiency (Krizhevsky et al., 2012). We tune the number of layers and kernel size for each modality.

3.3. Multi-mask Training

A disadvantage of the data2vec teacher-student setup is the need to process each sample twice: once to obtain targets with the teacher model, and once to obtain predictions of the student. Moreover, computing activations for the teacher model is also less efficient compared to the student model since the teacher needs to process the full unmasked input.[3]

In order to amortize the cost of the teacher model computation, we reuse the teacher representation for multiple masked versions of the training sample. Concretely, we consider M different masked versions of the training sample and compute the loss with respect to the same target representation. This is possible, because target representations are based on the full unmasked version of the sample. As M grows, the computational overhead of computing target representations becomes negligible. In practice, this enables training with a relatively small batch size compared to other self-supervised work (§4).

Considering multiple masked versions of a training sample has been previously explored in the context of self-supervised learning for computer vision with ResNet models (Jing et al., 2022), although the authors found that it performed much less well than different image augmentations. Caron et al. (2020a) considers multiple crops based on the same image but trains the model by comparing discrete codes rather than predicting the representation of the original image. And Girdhar et al. (2022) trains MAE models on videos with multiple masked versions of a sample to amortize the overhead of data loading and preparation.

Another efficiency improvement of data2vec 2.0 compared to data2vec is to share the feature encoder output across the different masked versions of the training example to avoid redundant computation. This leads to significant speedups for dense modalities such as speech where the feature encoder accounts for a large portion of the computation but less so for other modalities such as text.

3.4. Inverse Block Masking

The MAE-style sample encoding improves efficiency but also removes the ability to store information in the activations of masked time-steps which makes the training task more challenging. Random masking is successful for Masked Autoencoders (He et al., 2021) but it may interfere with the ability to build semantic representations since there is no structure in the masks that are created. Block masking (Bao et al., 2021) is more structured by masking entire blocks of time-steps or patches but there is no guarantee that large contiguous portions of the training sample are unmasked. Our goal is to enable the student model to build semantically rich representations over local regions of the sample.

We therefore introduce inverse block masking: instead of choosing which patches to mask, it chooses which patches to preserve in a block-wise fashion, where the size of a block is in terms of the number of patches or time-steps B. We first sample the starting point of each block to keep, and then expand it symmetrically until the block is of width B, for speech and text, or $\sqrt{B}$ for images.[4] We sample the following number of starting points without replacement and expand them to width B or quadratic blocks of width $\sqrt{B}$, depending on the modality:

$$L \times \frac{(1-R)+A}{B}$$

where L is the total number of time-steps/patches in a training sample, R is the mask ratio, a hyper parameter controlling the percentage of the sample that is masked and A is a hyper-parameter to adjust mask ratio (see below).

We allow blocks to overlap, which results in over-masking and some variance in the number of actually masked time-steps for each sample. Since we only encode unmasked time-steps, we use a simple strategy to assimilate the number of unmasked time-steps for all samples in a batch: for each sample, we randomly choose individual time-steps to mask or unmask until we reached the desired number of unmasked time-steps $L \times (1-R)$.[5]

4. Experiments

4.1. Efficiency

As a first experiment, we compare the efficiency of data2vec 2.0 pre-training to existing algorithms for vision, speech and NLP. We measure accuracy for image classification (§4.2), word error rate for speech recognition (§4.3), natural language understanding performance on GLUE (§4.4) and pre-training speed in terms of wall clock hours.

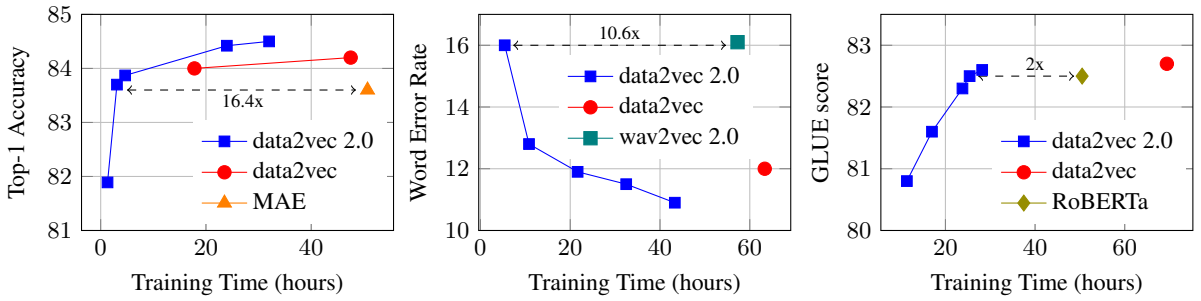
Setup. For computer vision, we compare to MAE (He et al., 2021) and data2vec (Baevski et al., 2020b) using their public implementations and recommended configurations.[6] Both data2vec 2.0 and data2vec are implemented

[3] Baevski et al. (2022) found it important to build targets based on the unmasked sample rather than another masked version.

[4] For speech and text the blocks are 1-D and block masking/inverse block masking perform similarly due to symmetry.

[5] We found $0.05 < A < 0.15$ to work well.

[6] data2vec: https://github.com/facebookresearch/fairseq/tree/main/examples/data2vec MAE: https://github.com/facebookresearch/mae/blob/main/PRETRAIN.md



(a) Computer Vision. (b) Speech Processing. (c) Natural Language Processing.

Figure 2. Efficiency of data2vec 2.0 for computer vision and speech processing in terms of wall clock time for pre-training Base models. Vision models are pre-trained on ImageNet-1K using 32 A100 40GB GPUs, then fine-tuned to perform image recognition and we report top-1 dev accuracy. Pre-training of speech models uses Librispeech and 16 A100 40GB GPUs, models are fine-tuned for speech recognition on the 10 hour labeled data split of Libri-light and we report word error rate on dev-other without a language model.

in fairseq (Ott et al., 2019) and we evaluate data2vec 2.0 configurations with different speed and accuracy trade-offs. All vision models are pre-trained on 32 A100 40GB GPUs, data2vec 2.0 models are pre-trained between 25k-500k updates, or 10-200 epochs, all with a total batch size of 512 images, $R = 0.8$, $M = 8$, except for the longest training run which uses $M = 16$. MAE is pre-trained for 500k updates using a batch size of 4,096 images or 1,600 epochs; data2vec is pre-trained for 500k updates with batch size 2,048 or 800 epochs.

Speech models are pre-trained on 32 A100 40GB GPUs, and data2vec 2.0 performs between 50k-400k updates, or 13-103 epochs, using a total batch size of 17 minutes of speech audio and we set $M = 8$, $R = 0.5$. We compare to wav2vec 2.0 and data2vec which are pre-trained for 400k updates with batch size 93min and 63min, respectively and following the recommended configurations. All models are implemented in fairseq.

NLP models are pre-trained on 16 A100 40GB GPUs, data2vec 2.0 uses between 400k-1m updates with a total batch size of 32 (each sample is a sequence of 512 tokens) and we set $M = 8$, $R = 0.42$. Models are compared to a retrained version of RoBERTa and data2vec are both pre-trained for 1m updates with a total batch size of 256 (32 epochs) following the original BERT setup. Models are implemented in fairseq.

Results. Figure 2 shows that data2vec 2.0 provides a far better speed and accuracy trade-off in all three modalities: an ImageNet pre-trained data2vec 2.0 model achieves a top-1 accuracy of 83.7% after pre-training for just over 3 hours vs. 83.6% after 50.7 hours for MAE - a 16.4x speed-up at slightly improved accuracy compared to the popular MAE algorithm (He et al., 2021). A speech data2vec 2.0 model achieves comparable word error rate to wav2vec 2.0 on speech recognition in 10.6x times lower wall clock time. For NLP, data2vec 2.0 trains to a similar accuracy as a retrained RoBERTa model in two times the speed.

The same models also perform far fewer epochs: for computer vision the data2vec 2.0 model with most similar accuracy to MAE performs 20 epochs vs. 1,600 epochs. For speech, data2vec 2.0 trains for 13 epochs vs. 522 epochs and for NLP, data2vec 2.0 performs four epochs compared to 32 for RoBERTa. data2vec 2.0 also provides a better efficiency compared to data2vec (Baevski et al., 2022): for vision, data2vec 2.0 can nearly match the accuracy of data2vec in 2.9x less time, for speech there is 3.8x speed-up, and for NLP there is 2.5x speed-up.

Note that data2vec is already faster than MAE: the most comparable data2vec model trained in 2.8x the speed of MAE (17.8 hours vs. 50.7 hours) - at higher accuracy (84.0% vs. 83.6%). Hence, the speed-up of data2vec 2.0 compared to MAE is much higher than for NLP, where the original data2vec was not more efficient than RoBERTa.

data2vec 2.0 can train well with a relatively small batch size of just 512 images, compared to 4,096 images in the case of MAE, or 2,048 images for data2vec and most other self-supervised algorithms for computer vision (§4.5 analyzes multi-masking in more detail). Training with a much lower number of epochs and batch size is possible because multi-masking extracts more learning signal from each training sample. Moreover, contextualized targets lead to a richer training task.

4.2. Computer Vision

Next, we compare data2vec 2.0 more broadly for each modality to existing work. For computer vision, we use a standard Vision Transformer architecture (Dosovitskiy et al., 2020) but with post-layer normalization, similar to the original Transformer architecture (Vaswani et al., 2017).

Table 1. Computer vision: top-1 validation accuracy on ImageNet-1K for ViT-B and ViT-L.

	epochs	ViT-B	ViT-L
Multiple models/external data			
BEiT (Bao et al., 2021)	800	83.2	85.2
PeCo (Dong et al., 2022)	800	84.5	86.5
BEiT-2 (Peng et al., 2022)	1600	85.5	87.3
TEC (Gao et al., 2022)	2400/ 1900	85.1	86.5
Single models			
MoCo-3 (Chen et al., 2021b)	300	83.2	84.1
DINO (Caron et al., 2021)	1600	82.8	-
MAE (He et al., 2021)	1600	83.6	85.9
SimMIM (Xie et al., 2021)	800	83.8	-
iBOT (Zhou et al., 2021)	1600	83.8	-
MaskFeat (Wei et al., 2021)	1600	84.0	85.7
data2vec (Baevski et al., 2022)	800/ 1600	84.2	86.6
data2vec 2.0	200/ 150	84.5	86.8

This results in an identical Transformer architecture for all modalities. We also apply random cropping and horizontal flipping to input images whose result and we feed the same augmented version both to the student and the teacher; we use the same hyper-parameters as MAE (He et al., 2021). For vision only, we found it useful to add a global CLS loss (Peng et al., 2022). For detailed hyper-parameters see Appendix A Table 8; for fine-tuning we use the same settings as He et al. (2021). We pre-train on the unlabeled version of ImageNet-1K.

Table 1 shows that data2vec 2.0 improves over prior single models using no external data both for ViT-B and ViT-L while training for far fewer epochs: compared to MAE, data2vec 2.0 increases accuracy by 0.9% while pre-training for less time (ViT-B: 32 hours vs. 50.7 hours, ViT-L: 63.3 hours vs. 93.3 hours). Compared to data2vec (Baevski et al., 2022) achieves slightly higher accuracy at far fewer epochs. data2vec 2.0 also improves over several approaches using multiple models and/or external data such as TEC (Gao et al., 2022), PeCo (Dong et al., 2022), and BEiT (Bao et al., 2021). BEiT-2 (Peng et al., 2022) performs better because it effectively distills representations from CLIP (Radford et al., 2021) which was trained on a much larger dataset than ImageNet-1K.

Table 2 shows the speed/accuracy trade-off for ViT-H models: data2vec 2.0 outperforms MAE by 0.5% while training for 40% less time and performing 1/16 of the number of training epochs.

Table 2. Computer vision: top-1 validation accuracy on ImageNet-1K for ViT-H/14. Pre-training time measurd on 64 A100 GPUs.

	epochs	ViT-H	Pre-train time (h)
MAE (He et al., 2021)	1600	86.9	113.6
data2vec 2.0	100	87.4	66.1

4.3. Speech Processing

To evaluate data2vec 2.0 on speech, we pretrain it on either Librispeech (Panayotov et al., 2015) or the much larger Libri-light dataset (Kahn et al., 2019) and fine-tune the resulting model for speech recognition on the labeled data splits of Libri-light which tests the model quality for different resource settings. See Table 9 in Appendix A for detailed hyper-parameters. We follow the fine-tuning regime of wav2vec 2.0 (Baevski et al., 2020b) whose hyper-parameters depend on the labeled data setup.

Alibi feature encoder. The feature encoder of Baevski et al. (2020b) uses relative positional embeddings modeled as a temporal convolution which assumes the that all time-steps are being encoded. We adapt this to our setup by removing parts of the kernel corresponding to masked time-steps. We also found it helpful to bias the query-key attention scores with a penalty proportional to their distance (Press et al., 2021). Biases are initialized following Press et al. (2021), but we keep them frozen during training and learn a scalar for each head which is initialized to 1.0. This adds very few new parameters (16 for a Large model), but leads to a significant improvement in accuracy which we ablate in §4.5.

The results (Table 3) show that data2vec 2.0 improves in most settings over prior work in less training time. Compared to wav2vec 2.0, data2vec 2.0 enables a relative word error rate reduction of up to 26% for Base models and up to 18% for Large models. For Base models, we use the most accurate model of Figure 2a which obtains higher accuracy than other models at faster training time (43.3 hours on 16 A100 40GB GPUs which as wav2vec 2.0 requires 57.3 hours on the same hardware). For Large models, we train data2vec 2.0 on 64 A100 40GB GPUs for 76.7 hours while as other models train for either 108 hours (data2vec) or 150 hours (wav2vec 2.0) on the same hardware.

4.4. Natural Language Processing

For NLP, we adopt the same training setup as BERT (Devlin et al., 2019) by pre-training on the Books Corpus (Zhu et al., 2015) and English Wikipedia using a 50k byte-pair encoding (Sennrich et al., 2016; Devlin et al., 2019; Liu et al.,

Table 3. Speech processing: word error rate on the Librispeech test-other when fine-tuning pre-trained models on the Libri-light low-resource labeled data setups (Kahn et al., 2020) of 10 min, 1 hour, 10 hours, the clean 100h subset of Librispeech and the full 960h of Librispeech. For pretraining, models use 960 hours of unlabeled audio from Librispeech (LS-960), or the 60K hours from Libri-light (LL-60K); WavLM Large uses 94K hours (MIX-94K) which includes LL-60K as well as other datasets. All results are based on 4-gram language models. We report wall clock pre-training time for Base models on 16 A100 GPUs and Large models on 64 A100 GPUs.

	Unlabeled data	LM	Amount of labeled data					Pre-train time (h)
			10m	1h	10h	100h	960h	
Base models								
wav2vec 2.0 (Baevski et al., 2020b)	LS-960	4-gram	15.6	11.3	9.5	8.0	6.1	57.3
HuBERT (Hsu et al., 2021)	LS-960	4-gram	15.3	11.3	9.4	8.1	-	-
WavLM (Chen et al., 2021a)	LS-960	4-gram	-	10.8	9.2	7.7	-	-
data2vec (Baevski et al., 2022)	LS-960	4-gram	12.3	9.1	8.1	6.8	5.5	63.3
data2vec 2.0	LS-960	4-gram	11.5	8.7	7.6	6.4	5.2	43.3
Large models								
wav2vec 2.0 (Baevski et al., 2020b)	LL-60K	4-gram	10.3	7.1	5.8	4.6	3.6	150.0
HuBERT (Hsu et al., 2021)	LL-60K	4-gram	10.1	6.8	5.5	4.5	3.7	-
WavLM (Chen et al., 2021a)	MIX-94K	4-gram	-	6.6	5.5	4.6	-	-
data2vec (Baevski et al., 2022)	LL-60K	4-gram	8.4	6.3	5.3	4.6	3.7	108.0
data2vec 2.0	LL-60K	4-gram	8.4	6.3	5.1	4.3	3.5	76.7

Table 4. Natural language processing: GLUE results on the dev set for single-task fine-tuning with Base models. For MNLI we report accuracy on the matched/unmatched dev sets, for MRPC and QQP, we report the unweighted average of accuracy and F1, for STS-B the unweighted average of Pearson and Spearman correlation, for CoLA Matthews correlation and accuracy for all other tasks. BERT Base results are from Wu et al. (2020), the baseline is a reproduction of BERT, pre-training time (PT) is measured on 16 A100 GPUs.

	epochs	MNLI	QNLI	RTE	MRPC	QQP	STS-B	CoLA	SST	Avg.	Pre-train time (h)
BERT	-	84.0/84.4	89.0	61.0	86.3	89.1	89.5	57.3	93.0	81.2	-
Baseline	31.8	84.1/83.9	90.4	69.3	89.0	89.3	88.9	56.8	92.3	82.5	50.5
data2vec	31.8	83.2/83.0	90.9	67.0	90.2	89.1	87.2	62.2	91.8	82.7	69.4
data2vec 2.0	4.1	83.7/83.7	90.7	68.6	88.8	89.3	87.3	59.1	92.9	82.6	28.2

2019). As baseline we retrain RoBERTa using the original BERT setup (Baseline; Liu et al. 2019) with the default BERT masking strategy (mask 15% of tokens) but without the next-sentence prediction task and we also compare to data2vec. Both RoBERTa and data2vec are pre-trained for 1m updates and with batch size 256. The hyper-parameters for data2vec 2.0 are in Appendix A Table 10.

Models are evaluated on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018) comprising tasks for natural language inference (MNLI, QNLI, RTE), sentence similarity (MRPC, QQP and STS-B), grammaticality (CoLA), and sentiment analysis (SST-2). Pre-trained models are fine-tuned on the labeled data provided by each task and we report the average accuracy on the development sets by performing nine different fine-tuning runs and reporting the average performance without the two best and the two worst performing runs to reduce the sensitivity to outliers.

The results (Table 4) shows that data2vec 2.0 achieves comparable average GLUE performance to our retrained RoBERTa baseline in 1.8x the speed and 7.8 fewer epochs. Compared to data2vec, there is a 2.5x speed-up. Note that data2vec 2.0 uses a much higher masking rate of 42% compared to 15% for BERT/RoBERTa which we believe is possible due to the use of rich contextualized targets.

4.5. Ablations

Multi-mask training. Next, we analyze the effect of multi-masking for different batch sizes. We use a reduced computer vision setup where we pre-train with 100k updates for a given batch size (bsz). Figure 3 shows that considering multiple masks per training sample can drastically improve accuracy, e.g., for bsz=64 considering $M = 16$ instead of $M = 2$ raises accuracy by 4.6% keeping everything else equal. This effect decreases with larger batch sizes but shows the possibility of pre-training high-quality models

Table 5. Training losses. ImageNet accuracy for removing the CLS loss, adding pixel regression (pixel regr), and only pixel regression. Results use a reduced setup (100 epochs).

	top-1 (%)
baseline	84.4
- cls loss	84.2
+ pixel regr	84.3
pixel regr only	83.5

Table 6. Masking strategy. Effect of block masking as well as different block sizes (B) for inverse block masking. $B = 1$ is equivalent to random masking and inv. block $B = 3$ is the default.

	top-1 (%)
block $B = 3$	84.1
inv. block $B = 1$	83.7
inv. block $B = 2$	84.4
inv. block $B = 3$	84.4
inv. block $B = 4$	84.2

Table 7. Alibi embeddings. WER on dev-other when removing Alibi, using only a single scalar for all heads and not learning scalars at all; pre-training is on LS-960 and fine-tuning with LL-10h.

	WER
baseline	10.9
- alibi	11.3
- learn scale/head	11.0
- learn scale	11.7

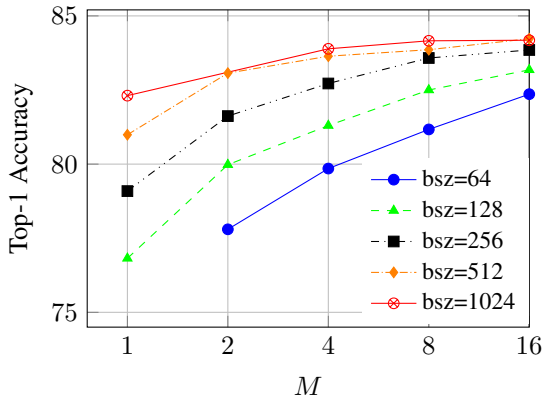


Figure 3. Multi-mask training (§3.3) enables pre-training with smaller batch sizes than usual. We show top-1 dev accuracy on ImageNet-1K for models pretrained with different batch sizes (bsz) and different number of masks per sample (M; §3.3); bsz=64 and $M = 1$ diverged due to too small overall batch size.

with dramatically lower batch size than is common today.

Training Losses. In the next experiment, we study our loss in more detail. Table 5 shows that the CLS loss component (§4.2) leads to a small improvement in accuracy for computer vision. The prediction of global representations as done by the CLS loss is complementary to predicting local patch information. We also compare contextualized target prediction to regressing the raw pixels of a local 16x16 patch of the training sample (He et al., 2021). Adding the MAE pixel regression loss (pixel regr) does not improve over contextualized target prediction alone and training only with the pixel regression loss (pixel regr only) results in a substantial drop in accuracy.

Masking Strategy. Next, we ablate our masking strategy by comparing it to block masking and random masking. Table 6 shows that block masking (block) performs less well than inverse block masking (our standard setting is $B = 3$); $B = 1$ corresponds to random masking and is also less effective.

Speech Alibi Embeddings. Finally, we investigate the effectiveness of the relative position embeddings for speech. Table 7 shows that the convolutional embeddings alone (baseline - alibi) perform less well than the alibi embeddings and that our design choices of learning scalars for the random embeddings are effective.

5. Conclusion and Future Work

We presented an efficient and general pre-training technique which relies on the same learning objective in different modalities. data2vec 2.0 shows that the training speed of self-supervised learning can be substantially improved with no loss in downstream task accuracy. At the heart of our approach lies the use of contextualized target representations with result in a more efficient self-supervised learner. Experiments show that data2vec 2.0 can reach the same accuracy as many popular existing algorithms in 2-16x the training speed. Future work includes the application of data2vec 2.0 to other modalities than vision, speech and text.

References

Akbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y., and Gong, B. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text, 2021.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. *arXiv*, abs/2204.14198, 2022.

Assran, M., Caron, M., Misra, I., Bojanowski, P., Bordes, F., Vincent, P., Joulin, A., Rabbat, M., and Ballas, N. Masked siamese networks for label-efficient learning. *arXiv*, abs/2204.07141, 2022.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv*, abs/1607.06450, 2016.

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., and Auli, M. Xls-r: Self-supervised cross-lingual speech representation learning at scale. In *Proc. of Interspeech*, 2022.

Baevski, A., Edunov, S., Liu, Y., Zettlemoyer, L., and Auli, M. Cloze-driven pretraining of self-attention networks. In *Proc. of EMNLP*, 2019.

Baevski, A., Schneider, S., and Auli, M. vq-wav2vec: Self-supervised learning of discrete speech representations. In *Proc. of ICLR*, 2020a.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proc. of NeurIPS*, 2020b.

Baevski, A., Hsu, W.-N., Conneau, A., and Auli, M. Unsupervised speech recognition. In *Proc. of NeurIPS*, 2021.

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., and Auli, M. data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv*, abs/2202.03555, 2022.

Bao, H., Dong, L., and Wei, F. Beit: BERT pre-training of image transformers. *arXiv*, abs/2106.08254, 2021.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Proc. of NeurIPS*, 2020.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv*, abs/2006.09882, 2020a.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv*, abs/2006.09882, 2020b.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. *arXiv*, abs/2104.14294, 2021.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., and Wei, F. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *arXiv*, abs/2110.13900, 2021a.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. *arXiv*, abs/2104.02057, 2021b.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. Palm: Scaling language modeling with pathways. *arXiv*, abs/2204.02311, 2022.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. URL `https://openreview.net/pdf?id=r1xMH1BtvB`.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proc. of NAACL*, 2019.

Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., Chen, D., Wen, F., and Yu, N. Peco: Perceptual codebook for bert pre-training of vision transformers, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, abs/2010.11929, 2020.

Eloff, R., Nortje, A., van Niekerk, B., Govender, A., Nortje, L., Pretorius, A., Van Biljon, E., van der Westhuizen, E., van Staden, L., and Kamper, H. Unsupervised acoustic unit discovery for speech synthesis using discrete latent-variable neural networks. *arXiv*, abs/1904.07556, 2019.

Gao, S., Zhou, P., Cheng, M.-M., and Yan, S. Towards sustainable self-supervised learning. *arXiv*, abs/2210.11016, 2022.

Girdhar, R., El-Nouby, A., Singh, M., Alwala, K. V., Joulin, A., and Misra, I. Omnimae: Single model masked pretraining on images and videos. *arXiv*, 2022.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv*, abs/2006.07733, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Proc. of CVPR*, 2015.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *arXiv*, abs/2111.06377, 2021.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv*, abs/1606.08415, 2016.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models. *arXiv*, 2022.

Hsu, W.-N., Tsai, Y.-H. H., Bolte, B., Salakhutdinov, R., and Mohamed, A. Hubert: How much can a bad teacher benefit ASR pre-training? In *Proc. of ICASSP*, 2021.

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., Hénaff, O., Botvinick, M. M., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver io: A general architecture for structured inputs & outputs. *arXiv*, abs/2107.14795, 2021a.

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., and Carreira, J. Perceiver: General perception with iterative attention. *arXiv*, abs/2103.03206, 2021b.

Jing, L., Zhu, J., and LeCun, Y. Masked siamese convnets. *arXiv*, abs/2206.07700, 2022.

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., Likhomanenko, T., Synnaeve, G., Joulin, A., Mohamed, A., and Dupoux, E. Libri-light: A benchmark for asr with limited or no supervision. *arXiv*, abs/1912.07875, 2019.

Kahn, J. et al. Libri-light: A benchmark for asr with limited or no supervision. In *Proc. of ICASSP*, 2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proc. of NIPS*, 2012.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv*, abs/1909.11942, 2019.

Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., and Gao, J. Efficient self-supervised vision transformers for representation learning. *arXiv*, abs/2106.09785, 2021.

Liu, A. T., Li, S.-W., and Lee, H.-y. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 2021.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL System Demonstrations*, 2019.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *Proc. of ICASSP*, pp. 5206–5210. IEEE, 2015.

Paulus, R., Xiong, C., and Socher, R. A deep reinforced model for abstractive summarization. *arXiv*, abs/1705.04304, 2017.

Peng, Z., Dong, L., Bao, H., Ye, Q., and Wei, F. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv*, abs/2208.06366, 2022.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv*, abs/2108.12409, 2021.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. `https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf`, 2018.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. *arXiv*, abs/2103.00020, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv*, abs/1910.10683, 2019.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. wav2vec: Unsupervised pre-training for speech recognition. In *Proc. of Interspeech*, 2019.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In *Proc. of ACL*, 2016.

Shi, B., Mohamed, A., and Hsu, W.-N. Learning lip-based audio-visual speaker embeddings with av-hubert. *arXiv*, abs/2205.07180, 2022.

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. FLAVA: A foundational language and vision alignment model. *arXiv*, abs/2112.04482, 2021.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. S. Instance normalization: The missing ingredient for fast stylization. *arXiv*, abs/1607.08022, 2016.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *Proc. of NIPS*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proc. of NIPS*, 2017.

Vyas, A., Hsu, W.-N., Auli, M., and Baevski, A. On-demand compute reduction with stochastic wav2vec 2.0. In *Proc. of Interspeech*, 2022.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv*, abs/1804.07461, 2018.

Wang, W., Bao, H., Dong, L., and Wei, F. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv*, abs/2111.02358, 2021.

Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. Masked feature prediction for self-supervised visual pre-training. *arXiv*, abs/2112.09133, 2021.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. Emergent abilities of large language models. *arXiv*, abs/2206.07682, 2022.

Wu, F., Kim, K., Pan, J., Han, K. J., Weinberger, K. Q., and Artzi, Y. Performance-efficiency trade-offs in unsupervised pre-training for speech recognition. In *Proc. of Interspeech*, 2022a.

Wu, Z., Wang, S., Gu, J., Khabsa, M., Sun, F., and Ma, H. CLEAR: contrastive learning for sentence representation. *arXiv*, abs/2012.15466, 2020.

Wu, Z., Lai, Z., Sun, X., and Lin, S. Extreme masking for learning instance and distributed visual representations. *arXiv*, 2206.04667, 2022b.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. *arXiv*, abs/2111.09886, 2021.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. ibot: Image bert pre-training with online tokenizer, 2021.

Zhu, Y., Kiros, R., Zemel, R. S., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *arXiv*, abs/1506.06724, 2015.

A. Pre-training Hyper-parameters

Table 8. Vision pre-training hyper-parameters. IN is instance normalization; AVG is mean pooling; LN is layer normalization.

	ViT-B	ViT-L	ViT-H/14
GPUs	32	32	32
Learning rate	1×10^{-3}	4×10^{-4}	4×10^{-4}
Adam β_1 / β_2	0.9 / 0.95	0.9 / 0.95	0.9 / 0.95
Weight decay	0.05	0.05	0.05
Clip norm	4.0	4.0	4.0
Learning rate schedule	cosine	cosine	cosine
Warmup updates	50,040	50,040	50,040
Batch size (per GPU / total)	16 / 512	8 / 256	8 / 256
Multi-masks (M)	16	16	16
CLS loss coefficient	0.01	0.01	0.01
τ_0 (EMA start)	0.9998	0.9998	0.9998
τ_e (EMA end)	0.99999	1.0	1.0
τ_n (EMA anneal steps)	100,000	500,000	300,000
B (block width)	3	3	3
R (mask ratio)	0.8	0.75	0.75
A (mask adjust)	0.07	0.1	0.1
K (layers to average)	10	18	32
Target normalization	IN $\rightarrow$ AVG $\rightarrow$ LN	IN $\rightarrow$ AVG $\rightarrow$ LN	IN $\rightarrow$ AVG $\rightarrow$ LN
Updates	500,000	750,000	500,000
Decoder dim.	768	1024	1024
Decoder conv. groups	16	16	16
Decoder kernel	3	5	5
Decoder layers (D)	6	3	3

Table 9. Speech pre-training hyper-parameters. IN is instance normalization; AVG is mean pooling.

	Base (Librispeech)	Large (Libri-light)
GPUs	16	64
Learning rate	7.5×10^{-4}	4×10^{-4}
Adam β_1 / β_2	0.9 / 0.98	0.9 / 0.98
Weight decay	0.01	0.01
Clip norm	-	1
Learning rate schedule	cosine	cosine
Warmup updates	8,000	10,000
Batch size (seconds per GPU / total)	62.5 / 1,000	20 / 960
Multi-masks (M)	8	12
τ_0 (EMA start)	0.999	0.9997
τ_e (EMA end)	0.99999	1.0
τ_n (EMA anneal steps)	75,000	300,000
B (block width)	5	5
R (mask ratio)	0.5	0.55
A (mask adjust)	0.05	0.1
K (layers to average)	8	16
Target normalization	IN $\rightarrow$ AVG	IN $\rightarrow$ AVG
Updates	400,000	600,000
Decoder dim.	384	768
Decoder conv. groups	16	16
Decoder kernel	7	7
Decoder layers (D)	4	4

Table 10. Natural language processing pre-training hyper-parameters. IN is instance normalization; AVG is mean pooling.

	Base
GPUs	16
Learning rate	2×10^{-4}
Adam β_1 / β_2	0.9 / 0.98
Weight decay	0.01
Clip norm	1.0
Learning rate schedule	cosine
Warmup updates	4,000
Batch size	32
Multi-masks (M)	8
τ_0 (EMA start)	0.9999
τ_e (EMA end)	1
τ_n (EMA anneal steps)	100,000
B (block width)	1
R (mask ratio)	0.42
A (mask adjust)	0
K (layers to average)	12
Target normalization	IN $\rightarrow$ AVG
Updates	1,000,000
Decoder dim.	768
Decoder conv. groups	1
Decoder kernel	9
Decoder layers (D)	5

B. Effect of Pre-training Dataset Size

Figure 4 shows the effect of randomly subsampling the pre-training data while keeping all hyper-parameters and the fine-tuning data constant. Increasing the amount of pre-training data helps larger models more, implying that the base size models underfit to ImageNet-1K with the data2vec style pre-training task.

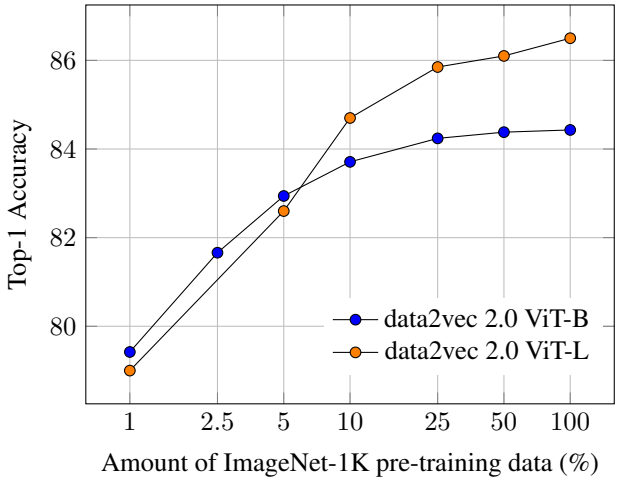


Figure 4. Top-1 accuracy when finetuning on the entire ImageNet-1K after pre-training on a subset of ImageNet-1K data.