# Self-Supervised Representation Learning: Introduction, advances, and challenges

# Self-Supervised Representation Learning: Introduction, Advances and Challenges

Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M. Hospedales



***Abstract*—Self-supervised representation learning methods aim to provide powerful deep feature learning without the requirement of large annotated datasets, thus alleviating the annotation bottleneck that is one of the main barriers to practical deployment of deep learning today. These methods have advanced rapidly in recent years, with their efficacy approaching and sometimes surpassing fully supervised pre-training alternatives across a variety of data modalities including image, video, sound, text and graphs. This article introduces this vibrant area including key concepts, the four main families of approach and associated state of the art, and how self-supervised methods are applied to diverse modalities of data. We further discuss practical considerations including workflows, representation transferability, and compute cost. Finally, we survey the major open challenges in the field that provide fertile ground for future work.**

## I. Introduction

DEEP neural networks (DNNs) now underpin state of the art artificial intelligence systems for analysis of diverse data types [1], [2]. However, the conventional paradigm has been to train these systems with supervised learning, where performance has grown roughly logarithmically with annotated dataset size [3]. The cost of such annotation has proven to be a scalability bottleneck for the continued advancement of state of the art performance, and a more fundamental barrier for deployment of DNNs in application areas where data and annotations are intrinsically rare, costly, dangerous, or time-consuming to collect.

This situation has motivated a wave of research in self-supervised representation learning (SSRL) [4], where freely available labels from carefully designed pretext tasks are used as supervision to discriminatively train deep representations. The resulting representations can then be re-used for training a DNN to solve a downstream task of interest using comparatively little task-specific annotated data compared to conventional supervised learning.

Self-supervision refers to learning tasks that ask a DNN to predict one part of the input data—or a label programmatically derivable thereof—given another part of the input. This is in contrast to supervised learning, which asks the DNN to predict a manually provided target output; and generative modelling, which asks a DNN to estimate the density of the input data or learn a generator for input data. Self-supervised algorithms differ primarily in their strategy for defining the derived labels to predict. This choice of *pretext task*, determines the (in)variances of the resulting learned representation; and thus how effective it is for different downstream tasks.

Self-supervised strategies have been leveraged successfully to improve sample efficiency of learning across a variety of modalities from image [5], [6], [7], video [8], [9], speech [10], [11], text [12], [13] and graphs [14], [15]. Across these modalities, it can also be applied to boost diverse downstream tasks including not only simple recognition but also detection and localisation [16], dense prediction (signal transformation) [16], anomaly detection [17], and so on. Furthermore, some results suggest that self-supervised representation quality is also a logarithmic function of the amount of unlabelled pre-training data [16]. If this trend holds, then achievable performance may improve for "free" over time as improvements in data collection and compute power allow increasingly large pre-training sets to be used without the need for manually annotating new data.

There are various other strategies for improving the data-efficiency of learning, such as transfer learning [18], [19], semi-supervised learning [20], active learning and meta-learning. As we shall see, SSRL provides an alternative competitor to conventional transfer learning and semi-supervised learning pipelines; however it can also be complementary to semi-supervised learning and active learning.

In this article, we focus on self-supervised algorithms and applications that address learning general-purpose features, or representations, that can be reused to improve learning in downstream tasks. We introduce self-supervised representation learning and review its application and state of the art across several modalities (image, text, speech, graphs, etc), with a specific focus on discriminative SSRL (we exclude generative models such as VAEs, GANs, and Flows; although they can also be used for representation learning). Compared to existing surveys [4], we provide a broader introduction to the field; a wider coverage of different modalities rather than focusing on images; highlight more practical considerations such as representation transferability, compute cost, and deployment strategies; and provide a deeper a discussion of open challenges.

## II. Background

### A. Problem Definition

In this section we introduce the necessary notation for defining the self-supervised representation learning problem and contrast it to other common learning paradigms (Figure 1).

*Supervised Learning* requires a labelled dataset for a target problem we wish to solve $D_t = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^N$, from which we build a predictive model that makes estimates $\hat{y} = f(x)$. In a deep learning context, the predictive model is usually composed of a representation extractor function $h_\theta$ and a classifier/regression function $g_\phi$, $f(x) = g_\phi(h_\theta(x)))$. We train

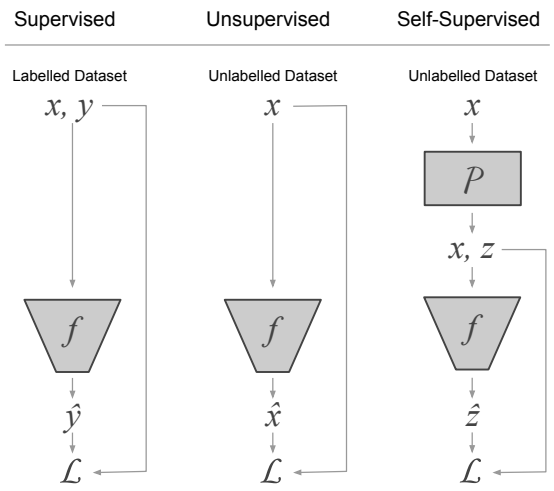


Fig. 1. Contrasting supervised, unsupervised and self-supervised learning paradigms for training a model $f$ using raw data $x$, labels $y$, and loss function $\mathcal{L}$. Self-supervision methods introduce pretext tasks $\mathcal{P}$ that generate pseudo-labels $z$ for discriminative training of $f$.

this predictive model by minimising a loss function $\mathcal{L}$ such as the negative log likelihood:

$$\arg\min_{\theta,\phi} \sum_{(x_i^{(t)},y_i^{(t)})\in D_t} \mathcal{L}(g_\phi(h_\theta(x_i^{(t)})), y_i^{(t)}). \quad (1)$$

However $h_\theta$ may have hundreds of millions of parameters, requiring millions of labelled data points in $D_t$ to fit this correctly. These millions of annotated data points are not available in most applications, but many do have an essentially free supply of *unlabelled* data points — as an example consider the wealth of raw audio signal data $x$ vs the limited amount of transcribed speech data $y$ in speech recognition.

*Unsupervised Learning* methods often learn from such unlabelled data by building generative models or density estimators. These range from classic shallow approaches such as Gaussian mixtures [21] to deep methods such as variational autoencoders (VAEs) and generative adversarial networks (GANs) [18]. Other common unsupervised approaches such as autoencoders and clustering [18] learn compact latent representations. For example autoencoders often optimise a reconstruction objective:

$$\arg\min_{\theta,\phi} \sum_{x_i^{(t)}\in D_t} \mathcal{L}(g_\phi(h_\theta(x_i^{(t)})), x_i^{(t)}), \quad (2)$$

where $h_\theta(\cdot)$ extracts a compact feature from the input, and $g\phi(\cdot)$ uses it to reconstruct the original input.

*Self-Supervised Representation Learning* can be seen as a special case of unsupervised learning, since both methods learn without annotations, $y$. While conventional unsupervised methods rely on reconstruction or density estimation objectives, SSRL methods rely on pretext tasks that exploit knowledge about the data modality used for training.

Although supervised learning methods tend to learn stronger features than unsupervised learning approaches, they require costly and time-consuming work from human annotators to generate the required labels. Self-supervised representation learning techniques aim for the best of both worlds: training a powerful feature extractor using discriminative learning, without the need for manual annotation of training examples.

Given an unlabelled *source* dataset $D_s = \{x_i^{(s)}\}_{i=1}^M$, with $M \gg N$, self-supervised learning addresses how to make use of $D_s$ and $D_t$ together to learn the predictive model $f(x) = g_\phi(h_\theta(x))$.

What defines a self-supervised method is its *pretext* task, consisting of a process, $\mathcal{P}$, to generate pseudo-labels and an objective to guide learning. Given a raw data set like $D_s$, the pretext process programmatically generates pseudo-labels $z$ and possibly modified data points $\{x_i, z_i\}_{i=1}^M = \mathcal{P}(D_s)$. As an example, a portion of a speech signal $x$ can be modified by masking out some part of the signal, and the pseudo-label $z$ is defined as the masked out portion of the input. A neural network can then be trained on the objective of predicting the missing portion $z$ given the partially masked $x$.

Most self-supervision research activity addresses deriving pretext tasks $\mathcal{P}$, which enable learning general purpose representations $h_\theta$ that provide high performance and data-efficient learning of downstream tasks $D_t$. Different pretext tasks are discussed in detail in Section III.

The workflow of self-supervision – also depicted in Fig. 2, proceeds as follows:

1) The annotated data for the target task forms the dataset $D_t$ and available unlabelled data forms the larger $D_s$.
2) The pretext task generates a new pseudo-labelled dataset as $\overline{D}_s = \{x_i, z_i\}_{i=1}^M = \mathcal{P}(D_s)$ as explained above. (As the process $\mathcal{P}$ often depends on sampling transformation or masking parameters, it is generally repeated at the start of each epoch of training.)
3) The pretext model, $k_\gamma(h_\theta(\cdot))$, is trained to optimise the self-supervised objective on $\overline{D}_s$,
$$\theta^* = \arg\min_{\theta,\gamma} \sum_{(x_i,z_i)\in\mathcal{P}(\overline{D}_s)} \mathcal{L}(k_\gamma(h_\theta(x_i)), z_i). \quad (3)$$
Importantly, this provides a good estimate $\theta^*$ of the potentially hundreds of millions of parameters in $h_\theta$, but without requiring label annotation. In many cases the input $x_i$ is a single datapoint and the pseudo-label $z_i$ is a class label of scalar value. However, as we will see later in certain types of instance discrimination methods, the input $x_i$ above can consist of multiple datapoints with the pseudo-label $z_i$ describing how the network should relate these datapoints. Similarly, in transformation prediction the input $x_i$ can consist of multiple shuffled chunks while the pseudo-label $z_i$ relates the shuffled order to the original order.
4) The pretext output function $k_\gamma$ is discarded, and the representation function $h_{\theta^*}$ is transferred as a partial solution to solve the target problem of interest using model $g_\phi(h_{\theta^*}(\cdot))$. Crucially, when representation parameters $\theta^*$ are already well fitted from the self-supervision step in Eq 3, only a minority of parameters may need to be learned or refined to solve the target problem, thus enabling it to be solved with a small labelled target dataset $D_t$. There are two common ways to solve the target problem using $\theta^*$ – fine-tuning and linear readout.

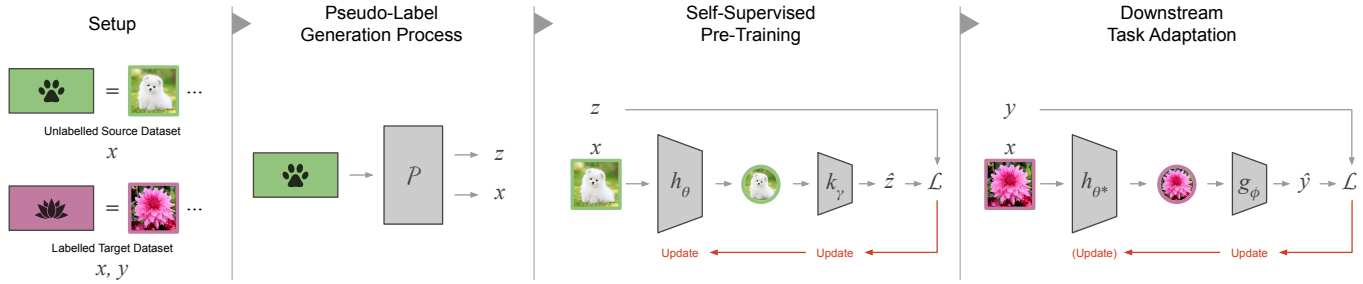


Fig. 2. The self-supervised workflow starts with an unlabelled source dataset and a labelled target dataset. As defined by the pretext task, pseudo-labels are programmatically generated from the unlabelled set. The resulting inputs, $x$ and pseudo-labels $z$ are used to pre-train the model $k_\gamma(h_\theta(\cdot))$ – composed of feature extractor $h_\theta$ and output $k_\gamma$ modules – to solve the pretext task. After pre-training is complete, the learned weights $\theta^*$ of the feature extractor $h_{\theta*}$ are transferred, and used together with a new output module $g_\phi$ to solve the downstream target task.

The presentation above assumes the target task is labelled and trained with supervised learning, as this is the most common use case. However, unlabelled target tasks like clustering or retrieval can also obviously benefit from self-supervised pre-training if substituted in step 4) above [22].

*Linear Readout* Let $(\theta, \gamma)$ be the weights of the pre-trained model consisting of a feature extractor $h_\theta$ followed by a task-specific head, $k_\gamma$. The simplest way to re-use $h_\theta$ for a new task is to replace the head with a new one, $g_\phi$, designed for the new task. This head is then trained with the feature extractor frozen. Given a target dataset of $N$ instances, $D_t = \{x_i^{(t)}, y_i^{(t)}\}_{i=1}^N$, the training objective is

$$\arg\min_{\phi} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(g_\phi(h_\theta(x_i^{(t)})), y_i^{(t)}). \tag{4}$$

The head is often a simple linear function, leading to the term linear readout. This is often used in the very sparse data regime where the number of unique parameters to learn for the target task must be aggressively limited to avoid over-fitting [23], [15], [24].

If enough downstream data is available, it may be better to fit a more complex non-linear function on top of the features. This may consist of multiple linear layers interspersed with non-linearities and potential task-specific modules. In the academic literature, however, it is very common to fit only linear functions in order to simplify comparison between methods.

*Fine-Tuning* Instead of just training a new head, we can retrain the entire network for the new task. We usually still need to replace the pretext head with one suited to the target task, but now we train both the feature extractor and the head,

$$\arg\min_{\theta,\phi} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(g_\phi(h_\theta(x_i^{(t)})), y_i^{(t)}). \tag{5}$$

Crucially, one must initialise $\theta$ with the values $\theta^*$ obtained during the self-supervised pre-training phase. Given that DNN optimisation is usually non-convex, and assuming a small learning rate, this results in the optimisation above converging towards a local optimum on the target task objective that lies near the local optimum attained for the source task, thus providing knowledge transfer from the pretext source task.

Fine-tuning is often used in the moderately sparse data regime where there is enough target data to at least refine all model parameters; or in regimes where the pretext task/data is not perfectly suited to the downstream task [5], [12], [25], [26], [27]. If the source and target domains are well aligned, it may not be necessary – or even beneficial – to fine-tune all the parameters. Often only the final few layers of a network need tuning in order to adapt to a new task. In other cases it is enough to tune a specific type of layer, like batch normalisation, to adapt to a slight change in domain.

In summary, self-supervised representation learning uses unlabelled data to generate pseudo-labels for learning a pretext task. The learned parameters then provide a basis for knowledge transfer to a target task of interest. After pre-training, the transfer can be completed by linear readout or finetuning on the labelled target data.

## B. Canonical Use Cases

When should one consider using self-supervision? SSRL may have diverse benefits in terms of adversarial robustness [28], model calibration [29], and interpretability [29] discussed further in Section VI. However its main use case is to improve data efficiency in situations where there are limited labels for the downstream target task (e.g., semantic segmentation or object detection) and/or domain (e.g., medical or earth observation images) of interest. We mention a few common problem templates and explain how SSRL fits in.

- If dense labels are available for the target task and domain, then direct supervised learning may be the most effective approach, and SSRL may not be helpful.
- If the target domain of interest is very different to any available background datasets (e.g., radar vs ImageNet data in imagery), and annotation is expensive in the target domain. Then collecting unlabelled target data for target-domain specific self-supervision, followed by sparse data fine-tuning may be effective. This setting can also be addressed by *semi-supervised* methods [20], which should then be evaluated as competitors against SSRL.
- If the target domain of interest is similar enough to large source datasets (e.g., everyday objects vs ImageNet).

Then one can leverage self-supervised pre-training on the source dataset before directly transferring the representation to the target domain of interest. Note that here *conventional supervised* pre-training is a competitor that should be evaluated against SSRL. However, in many cases state of the art SSRL has the edge on supervised pre-training for such transfer settings [29].

### C. Deployment Considerations

In this section we will discuss common ways of using a pre-trained encoder $h_\theta$ for a labelled target dataset. While there are often domain or task-specific methods in the literature for how to best do this, we will focus on some of the most widely adopted approaches.

The target input data is often assumed to lie in the same space as the source data, so that the encoder can be used without modification. The label spaces will most likely differ, however. This means that the head of the pre-trained network, $k_\gamma$ is not suited to solve the target task. The design of the new head $g_\phi$ depends mainly on the label space of the target task. For example, in object recognition, the output is likely a vector of class probabilities, for visual object detection additional bounding box locations must be predicted, and for dense prediction a deconvolutional decoder may be introduced.

*Layer Choice* Given the model pre-trained on the source task, which feature layer is best for extracting features in order to solve a downstream task is an active research question [16]. This is the problem of finding the correct layer to split the encoder $h_\theta$ and the source head $k_\gamma$. The optimal choice can differ from task to task and dataset to dataset, and can involve combining features from several layers, but a general rule is that earlier layers tend to encode simple patterns while later layers can combine these simpler patterns into more complex and abstract representations.

*Fine-tuning vs Fixed Extractor* An important design choice in deployment phase is whether to fix encoder $h_\theta$, and just train a new classifier module $g_\phi$ using the target data, or fine-tune the encoder while training the classifier. Many SSRL benchmarks use an experimental design relying on linear classifier readout of a frozen encoder. This makes SSRL methods easier to compare due to there being fewer parameters to tune in linear readout.

There have been mixed results reported in the literature with regards to whether linear readout is sufficient, or whether fine-tuning the entire encoder should improve performance [30], [29]. Which performs better may depend on the amount of available data (fine-tuning is more reliable with more data), the similarity between source and target domain data, and how well suited the (in)variances of the SSRL pretext task used are to the requirements of the downstream task. Conditions with larger domain/task discrepancy are likely to benefit from more fine-tuning. Of course there are numerous ways to control the amount of fine-tuning allowed in terms of learning rate, and explicitly regularising the fine-tuning step to prevent it over-fitting by limiting the deviation from the initial pre-trained conditions [19].

*Other Considerations* A unique issue for SSRL is that it can be difficult to determine the ideal stopping condition for the pretext task as no simple validation signal that can be used. There is not yet an efficient solution for this issue.

Multiple studies have observed that downstream performance after SSRL improves with the capacity of the network architecture used for pre-training [16], [5]. While convenient for extracting more performance at the cost of compute and memory, it may create a bottleneck for deploying the resulting fat representation on an embedded or other memory-constrained downstream platform. To alleviate this issue, high-performance and high-parameter count SSRL features can be *distilled* into smaller networks while retaining their good performance [5], [31] (see also Section VI-C).

## III. Pretext Tasks

In the absence of human-annotated labels, self-supervision uses the intrinsic structure of the raw data and an automated process $\mathcal{P}$ to synthesise a labelled source dataset, $\overline{D}_s = \{x_i, y_i\} = \mathcal{P}(D_s)$. One can then make use of $\overline{D}_s$ as they would any other labelled dataset when pre-training a model, by applying a discriminative supervised learning algorithm. As the pseudo-labels are created from some intrinsic structure in the data, a model learning to predict those labels must recognise and exploit this structure to solve the task successfully. Thus self-supervised algorithm design requires and exploits human prior knowledge about structure in the data to help define meaningful pretext tasks. Furthermore, different pretext tasks will induce different (in)variance properties in the learned representations, so the choice of method can also be informed by what properties of the representation are required by the downstream task. In this section we divide the various self-supervised pretexts in the literature in four broad families of *masked prediction*, *transformation prediction*, *instance discrimination*, and *clustering* – as illustrated in Figure 3.

### A. Masked Prediction

This family of methods is characterised by training the model to *fill in missing data* removed by $\mathcal{P}$. It relies on the assumption that context can be used to infer some types of missing information in the data if the domain is well modelled. Given a raw example, $x_i^{(s)}$, a subset of the elements are extracted to form the pseudo-label, $z^i$, and the remaining components that were not used to create the label are used as the new input example, $x_i$. The pseudo-label generation process therefore looks like $x_i, z_i = \mathcal{P}(x_i^{(s)})$ and is described in full in Alg. 1.

As an example of this on real data, a square region of an image can be masked out in the raw example. In this scenario $I$ is the set of indices inside the square mask region, the pixels in the masked region will correspond to $z_i$, and the pixels outside the masked region will be $x_i$. Given $x_i, z_i$, the model can now be trained to minimise e.g., a reconstruction loss like mean squared error,

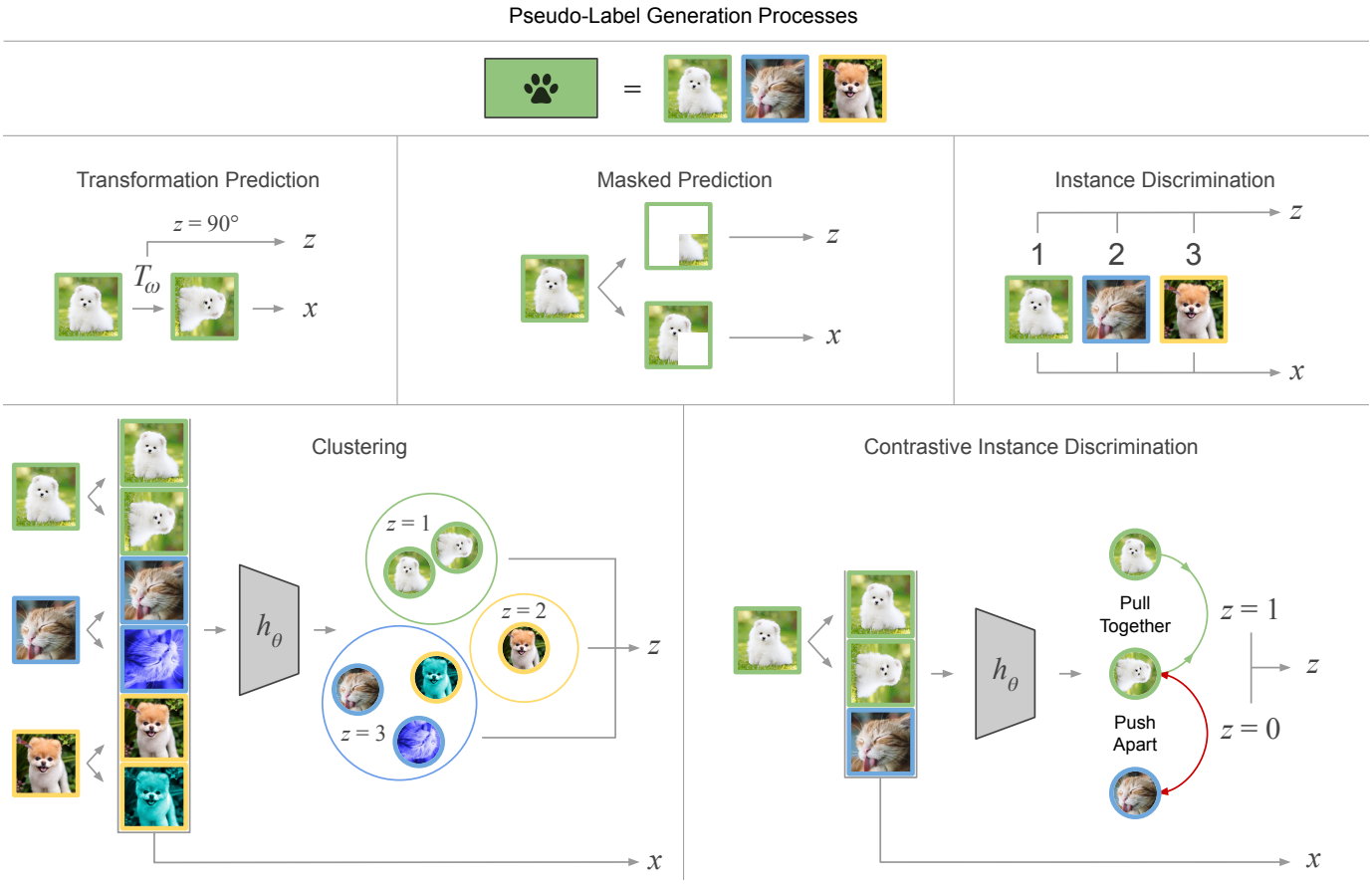


Fig. 3. Illustrative examples of the way pseudo-labels are generated in the four families of pretext tasks of our taxonomy: transformation prediction, masked prediction, instance discrimination and clustering. An additional depiction is included of the popular version of instance discrimination using contrastive losses. Squares represent inputs $x$ while circles portray the feature vectors of those inputs, $h_\theta(x)$.

**Algorithm 1** Pseudo-label generation process $\mathcal{P}$ for masked prediction

**Input:** Unlabelled dataset $D_s = \{x_i^{(s)}\}_{i=1}^M$.
  **for** $i$ from 1 to $M$ **do**
    Generate indices, $I$, of elements to remove from $x_i^{(s)}$
    $z_i \leftarrow \{x_{i,j}^{(s)} : j \in I\}$
    $x_i \leftarrow \{x_{i,j}^{(s)} : j \notin I\}$
  **end for**
**Output:** $\{x_i, z_i\}_{i=1}^M$.

$$\theta^* = \arg\min_{\theta,\gamma} \frac{1}{|\mathcal{P}(D_s)|} \sum_{(x_i,z_i)\in\mathcal{P}(D_s)} \Big(k_\gamma(h_\theta(x_i)) - z_i\Big)^2. \quad (6)$$

A major variant of masked prediction approaches are auto-regressive methods, which treat $x$ as a sequence, and the task is to auto-regressively predict the $t+1$ element of the sequence given the $t$ elements seen so far. By factorizing the joint distribution over $x$ into a product of conditionals, these methods can also be seen as unsupervised generative models.

*Examples* Common masking methods involve hiding words in sentences for language modelling [22], [12], [32], hiding time-slices in speech [10], hiding regions of images for inpainting [26], or hiding edges in graphs [27]. In a multi-modal setting it could correspond to, e.g., predicting the audio signal accompanying a video input or vice-versa.

*Considerations* Defining an ideal masking strategy (how much, when and where to mask; which context to provide in predicting the masked information) is important in making effective use of masked prediction. For example, masking too much of a speech signal will make it impossible to infer the missing words, while masking too little of it makes the task too easy to require a rich speech model to be learned.

### B. Transformation Prediction

This family relies on the assumption that inputs have a canonical view and that certain transformations can be applied to that view to change it. The canonical view can for example depend on the effects of gravity in vision (i.e., there is a correct notion of up and down in visual scenes) or temporal ordering in video, speech, or other time-series. Transformation prediction methods apply a transformation that maps from canonical views to alternative views and train the model to predict what transformation has been applied. Given a raw input in its canonical view, $x_i^{(s)}$, a transformation $T_\omega$ is applied to produce $x_i = T_\omega(x_i^{(s)})$, which is fed into the model.

**Algorithm 2** Pseudo-label generation process $\mathcal{P}$ for transformation prediction

**Input:** Unlabelled dataset $D_s = \{x_i^{(s)}\}_{i=1}^M$.
  **for** $i$ from 1 to $M$ **do**
    Sample $\omega \sim \Omega$
    $x_i \leftarrow T_\omega(x_i^{(s)})$ ▷ Apply transformation to raw input
    $z_i \leftarrow \omega$
  **end for**
**Output:** $\{x_i, z_i\}_{i=1}^M$.

The parameters, $\omega$, of this transform are used as the pseudo-label, $z_i = \omega$, that the model is trained to predict. It is typical for these transformation parameters to be sampled from some distribution, $\Omega$. The learning objective can be, e.g., a cross-entropy loss in the case of categorical transformation parameters.

$$\theta^* = \arg\min_{\theta,\gamma} \sum_{(x_i,z_i)\in\mathcal{P}(D_s)} \mathcal{L}_{CE}\Big(k_\gamma(h_\theta(x_i)), z_i\Big). \quad (7)$$

The full process $\mathcal{P}(D_s)$ is described in Alg. 2. Typically one will generate several different views of each $x_i^{(s)}$, each with a different set of transformation parameters. To succeed, a SSRL method has to learn enough about the latent structure of the data to correctly predict the transformation while being invariant to intra-category variability.

*Examples* In vision applications, one can apply rotations to the raw images and requiring the network to predict angle of rotation [33]. In temporal data, such as videos and other time-series, one can shuffle the temporal order of signal samples, and force the network to predict the original order [8], [34].

*Considerations* Whatever transformation is chosen, the model will learn to produce representations that are equivariant to that transformation. This is because the information regarding the transformation needs to be retained in the representation in order for the final layer to be able to correctly solve the pretext task. A second consideration is it that depends on data having a canonical view. If there is no canonical view with respect to the set of transformations, then performance will be poor. E.g., satellite or drone earth observation image data may have no canonical view with respect to rotation, so training for rotation prediction on this data may be ineffective.

### C. Instance Discrimination

In this family of methods, each instance in the raw source dataset, $D_s$, is treated as its own class, and the model is trained to discriminate between different instances. There are a few different variations on this framework which we will now describe.

*Cross-entropy* The most straightforward way of tackling instance discrimination is to assign each instance in the dataset a one-hot encoding of its class label, e.g. instance number 126 in a dataset of 100,000 images would be assigned a vector of length 100,000 with zeros everywhere except for a value of one at position 126. This enables training the network with a categorical cross-entropy loss to predict the correct instances. This was the approach taken by the early Exemplar-CNN method [23]. However, as the size of the dataset grows, the softmax operation to compute class probabilities becomes prohibitively expensive. As such it became difficult to scale this method to large modern datasets where the number of instances—and therefore classes—can be millions [35] or even billions [36]. This led to the development of contrastive methods discussed below.

Another problem within the instance discrimination framework is the lack of intra-class variability. Since each instance in the dataset is treated as its own class, we end up with only a single example of each class. In conventional supervised learning there might be hundreds or thousands of examples within each class to aid the network to learn the inherent variation within in each class. This problem was tackled by Exemplar-CNN via extensive data augmentation. Given a datapoint, we can apply many different transformations to obtain slightly different views of that same datapoint, while preserving its core semantic information. For example, we can slightly change the colour of an image of a car, and it will still be perceived as an image of a car. Figure 4 shows examples of common transformations across the modalities. The use of data augmentation has become an important component for instance discrimination methods as we will see in the more recent contrastive and regularisation-based methods discussed next.

*Contrastive* The issue with using a categorical cross-entropy loss to solve instance discrimination is that it becomes intractable for large datasets. Researchers therefore looked for ways to approximate this loss in more efficient ways. The core idea leading recent advances is inspired by metric learning, as well as [37] and [38]. The idea is to not predict the exact class of the input but to instead predict whether pairs of inputs belong to the same or different classes. This allows the use a binary class label instead of massively high-dimensional class vectors. If a pair of inputs belong to the same class the label is one and if they belong to different classes the label is zero. In this setting, however, the use of data augmentation becomes even more important, as we need to introduce variation between inputs of the same class.

To formalise the contrastive instance discrimination setup, multiple views of inputs are created via some process $T$ (transformation or sensory-based) and compared in representation space. One input, $x^a \sim T(x_i^{(s)})$, is chosen to be the *anchor*, and is compared with a positive sample, $x^+ \sim T(x_i^{(s)})$, which is another view or transform of the same input. The anchor is also contrasted with a negative sample, which is a view of a *different* image, $x_j^- \sim T(x_j^{(s)})$. In the context of the general SSRL objective given in Eq. 3, this means that the pretext task generator $\mathcal{P}$ produces pretext inputs that each correspond to multiple *pairs* of raw input instances, with associated pseudo-labels indicating whether the pairs are matching or mismatching. See Alg. 3 for a full description.

The samples are then encoded by the feature extractor to

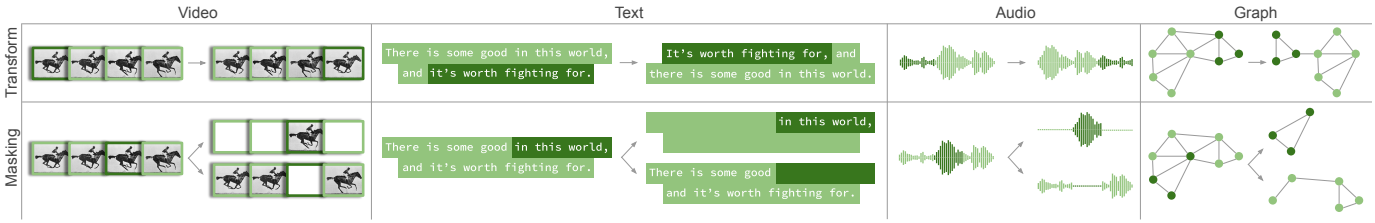


Fig. 4. Common transformation and masking methods for different modalities of data. Transformations can be applied to alter the order of sequential data, like the frames in a video, clauses in text or chunks in audio waves. Graphs can be transformed by moving nodes or neighbourhoods. Masking can be applied by hiding frames in videos, groups of words in text, chunks for audio data or subgraphs in graphs. The darker green highlights the portion of the data point that is transformed or masked out. For examples of transforms and masking on image data, see Figure 3.

**Algorithm 3** Pseudo-label generation process $\mathcal{P}$ for contrastive instance discrimination

**Input:** Unlabelled dataset $D_s = \{x_i^{(s)}\}_{i=1}^M$.
  **for** $i$ from 1 to $M$ **do**
    Sample $x^a \sim T(x_i^{(s)})$
    Sample $x^+ \sim T(x_i^{(s)})$
    **for** $k$ from 1 to $K$ **do**
      Sample $j \sim \mathcal{U}(1, M)$ ▷ Pick another raw input.
      Sample $x_k^- \sim T(x_j^{(s)})$ ▷ Get a random transform
    **end for**
    $x_i \leftarrow \{(x^a, x^+), (x^a, x_1^-), ..., (x^a, x_K^-)\}$.
    $z_i \leftarrow \{1, 0, ..., 0\}$.
  **end for**
**Output:** $\{x_i, z_i\}_{i=1}^M$.

obtain their representations, $r^a = h_\theta(x^a)$, $r^+ = h_\theta(x^+)$, $r_j^- = h_\theta(x_j^-)$. A similarity function $\Phi$ is used to measure the similarity between positive pairs (the anchor with a positive sample) and negative pairs (the anchor with a negative sample). The system is then trained to pull positive pairs closer and push negative pairs apart. A general formulation of the contrastive loss used in many works is

$$\mathcal{L}_{con} = -\mathbb{E}\left[\log \frac{\Phi(r^a, r^+)}{\Phi(r^a, r^+) + \sum_{j=1}^k \Phi(r^a, r_j^-)}\right], \quad (8)$$

where $k$ different negative samples have been contrasted with the anchor. The model can then be updated by minimising the contrastive loss

$$\theta^*, \gamma = \arg\min_{\theta,\gamma} \sum_{(x_i, z_i) \in \mathcal{P}(D_s)} \mathcal{L}_{con}\Big(k_\gamma(h_\theta(x_i)), z_i\Big). \quad (9)$$

Within this framework, methods differ in what similarity function they use, whether they use the same or different encoders for the anchor and other samples, what family of transformations $T$ they use, and how they sample anchor, positive and negative examples. Notable contrastive instance discrimination methods are SimCLR [5] and DGI [15].

*Regularisation-based* While the contrastive framework succeeds in scaling instance discrimination to large datasets, it still has some issues. In order to learn efficiently a very large number of negative examples need to be included in the loss. If we use too few negative examples the network will fail to learn the subtle differences between instances, but too many and training will be computationally expensive. If we were to remove negative examples altogether the features of our network would all collapse to a single constant vector, as there is no incentive to separate features.

Regularisation-based approaches to instance discrimination avoid the use of negative examples altogether by regularisation techniques that prevent feature collapse while keeping training efficient. There are many different techniques like using asymmetrical encoding for the two inputs [6] or minimising redundancy via the cross-correlation between features [39].

*Examples* Established SSRL methods for computer vision including MoCo [40] and SimCLR [5] fall into this family. Other applications include speech [11]; and multi-view [41] and multi-modal representation learning including audio-visual [42] and visuo-linguistic [43] data – where matching and mismatching views of the same instance are contrasted against each other.

*Considerations* The representations learned here develop high sensitivity to instances, while developing invariance to transformations or views. This means the design of augmentation or view-selection function $T$ is important due to its influence on the invariances learned. For example, aggressive colour augmentation in $T$ may lead to colour invariant representations [29], which could either be an issue or a benefit depending on the downstream task. If using different speakers as different views for audio data, the representations would become speaker invariant, which could be beneficial if the downstream task is speech recognition, but an issue if it is speaker diarisation.

Recent work has systematically demonstrated this intuition that the ideal transformations to use do indeed depend on the downstream task [44]. On one hand this undermines the appealing and widely believed property of SSRL that a single pre-trained model can be re-used for diverse downstream tasks. On the other hand it highlights a new route for research to further improve performance by customising the transformation choice according to the downstream task requirements.

Instance discrimination methods implicitly assume that all instances in the raw dataset represent unique semantic examples, which might not hold – e.g., if there are many images of the same object. When this assumption is violated, they suffer from *false-positive* pretext task labels [45]. Nevertheless, they are highly effective in practice despite this violated

assumption.

A different issue that is not well understood in theory, but crucial in practice is the sampling and batching strategy for anchor, positive and negative instances for contrastive methods. For example: How to choose negative samples (e.g., at random, via hard negative mining)? What proportion of positive and negative samples, and the size of batches to use [40]? These are all crucial design parameters that vary across the many methods and significantly influence performance.

### D. Clustering

This family of methods focuses on dividing the training data into a number of groups with high intra-group similarity and low inter-group similarity. This relies on the assumption that there exists meaningful similarities by which the data can be grouped – which is likely the case especially if the data is categorical in nature. There are multiple ways of determining cluster assignment such as connectivity (hierarchical clustering), centroids fitting (e.g. $k$-means), likelihood maximisation (e.g. Gaussian mixture modelling) and more [21].

As opposed to traditional clustering, in self-supervised representation learning the aim of the algorithm is to obtain a good feature extractor $f_\theta$ instead of the cluster assignments. Thus one typically jointly performs feature extractor learning and clustering to pre-train the representation prior to downstream use. This is in contrast to classic clustering methods which usually use a fixed set of features.

A common approach to self-supervised clustering is by alternating two steps, (1) *optimising the clustering objective* by assigning datapoints into clusters based on their representations and (2) *optimising the model* by using the cluster assignments as the pseudo-labels in updates.

A unique feature of the clustering family is thus that the pretext task $\mathcal{P}$ changes during the course of training. Since the pseudo-labels are created by clustering the current representations at each epoch, the labels are updated as the representations change. This means that the input to the process $\mathcal{P}$ at each iteration is the representations and clusters in addition of the raw data. The full process $\mathcal{P}$ is described in Alg. 4.

**Algorithm 4** Pseudo-label generation process $\mathcal{P}$ for clustering

**Input:** Unlabelled dataset $D_s = \{x_i^{(s)}\}_{i=1}^M$.
**Input:** Representations $\{r_i\}_{i=1}^M$, where $r_i \leftarrow h_\theta(x_i^{(s)})$
**Input:** Cluster centres $\{c_j\}_{j=1}^k$, via clustering on $\{r_i\}_{i=1}^M$.
  **for** $i$ from 1 to $M$ **do**
    Sample $x_i \sim T(x_i^{(s)})$
    $z_i \leftarrow \arg\min_{j\in[k]} \|c_j - r_i\|$
  **end for**
**Output:** $\{x_i, z_i\}_{i=1}^M$.

Given a cluster assignment, where each input, $x_i$, has its cluster class assigned to $z_i$, we can optimise the model via a cross-entropy loss

$$\theta^* = \arg\min_{\theta,\gamma} \sum_{(x_i,z_i)\in\mathcal{P}(D_s)} \mathcal{L}_{CE}\Big(k_\gamma(h_\theta(x_i)), z_i\Big). \quad (10)$$

After this, we go back to the clustering step, now using the new representations of our updated model.

In the cluster assignment step, many works use $k$-means clustering [46], [7], where the number of clusters $k$ is a hyperparameter set by evaluating on a validation set of a downstream task. A big problem is that there are degenerate solutions to this, such as assigning all instances to the same cluster [46]. To avoid this, methods often enforce that clusters assignments must be balanced [7]. Recent approaches such as ODC [47] aim to avoid the burden of alternating updates of feature extractor and clusters by simultaneously updating both online.

*Examples* Major examples include DeepCluster, ODC [46], [47] and SwAV [7] for vision; and XDC [9] for multi-modal clustering such as audio and video.

*Considerations* Many clustering-based SSRL methods [46] rely less heavily on data augmentation compared to contrastive methods [5]. This, and avoiding the need to sample triplets, has some benefit in terms of compute cost, but on the other hand the non-stationary nature of the clustering SSRL task (clusters co-evolve with features) imposes additional cost compared to the other pretext tasks with stationary objectives.

Compared to instance discrimination, transformation prediction and masked prediction pretexts, it can be harder to analyse the kinds of (in)variances induced by clustering-based SSRL, making it harder to predict which downstream tasks they are suitable for without empirical evaluation.

## IV. Theoretical Underpinning

The theoretical underpinnings of self-supervised representation learning are lacking compared to standard supervised learning. When analysing a conventional supervised method, the object of most interest is the expected performance of a model on unseen data. The model performance is measured using a task-specific loss function. For example, consider the case of a binary classification problem where the model produces a real-valued score; the sign of this score indicates the predicted class, and the magnitude provides an indication of the confidence with which the model is making the prediction. One loss function that is commonly used for evaluation purposes is the zero–one error,

$$\mathcal{L}_{0-1}(f, x, y) = \mathbb{I}(f(x)y > 0), \quad (11)$$

where $y \in \{-1, 1\}$ is the ground truth label, $f$ is the model, $x$ is an input, and $\mathbb{I}(\cdot)$ is the indicator function. The expected performance of a model on unseen data is then denoted by

$$\mathbb{E}_{x,y}[\mathcal{L}_{0-1}(f, x, y)]. \quad (12)$$

The typical goal in statistical learning theory is to bound this quantity from above using the error measured on the training set and some measure of complexity of the class of models, $\mathcal{F}$, that the training algorithm is optimising over. Such bounds are probabilistic, due to the inherent randomness involved in sampling a training dataset, and in some sense can be thought

of as sophisticated confidence intervals. These bounds hold uniformly over all $f \in \mathcal{F}$, and usually take the form

$$\mathbb{E}_{x,y}[\mathcal{L}_{0-1}(f,x,y)] \leq \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_{0-1}(f,x_i,y_i) + \mathcal{C}(\mathcal{F},n,\delta), \tag{13}$$

where the inequality holds with a probability of at least $1-\delta$, so $\delta$ is essentially defining the width of a confidence interval as in classic statistical analysis. The complexity term, $\mathcal{C}(\mathcal{F},n,\delta)$, can be thought of as the upper bound of a confidence interval that takes into account multiple hypothesis testing—i.e., each $f \in \mathcal{F}$ can be thought of as a hypothesis. As more complex classes of models are considered this term will grow larger. Crucially, these bounds assume no knowledge about the underlying data generating distribution, and as such they hold for all distributions.

There are several roadblocks preventing the direct application of this framework to SSRL methods. The most fundamental issue is that the training loss used during self-supervised pre-training measures performance on a pretext task, and is generally not the same loss function used for measuring the performance of the downstream task. As a consequence, the training loss cannot be interpreted as a biased estimate of the expected model performance, and analysis of the model class complexity cannot be used to compensate for the bias in this estimate by widening the confidence interval. A further complication comes from distribution shift. In many cases one wishes to perform self-supervised pre-training on one dataset (such as ImageNet), and then use the resulting features on another dataset with a different marginal distribution. One of the standard assumptions made in learning-theoretic analysis is that elements in the training set and the test set are sampled from the same distribution.

Nevertheless, there is a small but growing literature concerned with theoretical analysis of self-supervised representation learning methods. The key goal these papers share is relating a self-supervised training objective to a supervised objective measured on a small set of labelled data by, e.g., showing that the SSRL loss can be interpreted as an upper bound to a supervised loss. Such analyses typically rely on making assumptions about the data generating process that are hard to verify in practice. We will briefly outline three recent approaches to connecting SSRL with conventional statistical learning theory: one method that applies only to instance discrimination methods [48], and another that primarily considers how SSRL learns useful representations for natural language tasks [49], and finally a paper that makes use of conditional independence to further elucidate how masked prediction pretext tasks lead to useful representations.

The analysis of contrastive instance discrimination methods for self-supervised representation learning [48] is predicated on the assumption of a specific data generating process. In particular, they assume that the data is generated by a mixture of distributions associated with latent classes. E.g., there is a distribution over the pixels in an image associated with the concept 'dog', and there is some prior probability that an image from a particular domain will contain a dog. They demonstrate that one can bound the supervised loss by

$$\mathbb{E}_{x,y}[\mathcal{L}_{0-1}(f,x,y)] \leq \mathbb{E}_{x}[\mathcal{L}^{-}_{ssrl}(f,x)] + s(\rho) + \mathcal{C}(\mathcal{F},n,\delta), \tag{14}$$

where $\mathcal{L}^{-}_{ssrl}(\cdot,\cdot)$ is a modification to the contrastive loss that considers only negative pairs, and $s(\rho)$ is a function of the mixing coefficients, $\rho$, over the latent classes. This bound relies on $f$ being a centroid classifier on top of the network trained with SSRL, and it is shown that this line of analysis is of limited use on more general families of models.

SSRL on text data is often formalised as a masked prediction problem where, given the first part of a sentence, the task is to predict the next word or remainder of the sentence. Recent work [49] has provided a concrete link between the performance on this pretext task and the performance one can expect to see on natural language classification problems. However, their analysis does require an assumption for how classification tasks can be reformulated to make them more comparable with the sentence reconstruction pretext task. Their first contribution is to formalise this assumption as a falsifiable hypothesis and empirically verify that it holds in practice. Their second main contribution investigates the transfer performance of $\epsilon$-optimal language models—models that achieve an expected cross entropy loss within $\epsilon$ of the expected loss of the best possible model. They show that, conditioned on this empirically verified hypothesis being true, if one can find a model for next word prediction with an $\epsilon$-optimal cross entropy loss, then the cross entropy loss for a downstream classification task will be $O(\sqrt{\epsilon})$. This implies that developing models that are better at the next word prediction pretext task will translate into better feature representations for natural language classifiers.

Lee et al. [50] conduct a more general analysis of masked prediction pre-text tasks that is not restricted specifically to the natural language processing domain. Recall that masked prediction pre-text tasks take each source instance, $x_i^{(s)}$, and produce two new objects, $x_i$ and $z_i$, which contain subsets of the elements in the original instance. It is shown in [50] that if there is conditional independence between $x_i$ and $z_i$ given the downstream label (and optionally some additional latent variables), then any model that successfully predicts $z_i$ from $x_s$ must be estimating the label (and optional latent variables). They further generalise their results to the case where one must only assume some notion of approximate conditional independence, which they quantify in terms of covariance matrix norms.

While there has been some advances in understanding why contrastive and masked prediction methods can lead to discriminative representations for downstream tasks, this work does rely on assumptions about the data (e.g., conditional independence) that has not been verified to occur in practice. Moreover, the empirical results associated with methods from other parts of our taxonomy, such as transformation prediction and deep clustering, have still not been investigated. An example of how further work could address gaps in our current understanding is to extend theoretical frameworks analysing (shallow) clustering methods [51] to the deep SSRL paradigm. Future work addressing these limitations would be useful for

TABLE I
NOTABLE METHODS IN EACH MODALITY. CODE/PT: INDICATES WHETHER A CODE-BASE AND PRE-TRAINED MODELS ARE AVAILABLE, RESPECTIVELY. TP: TRANSFORMATION PREDICTION. MP: MASKED PREDICTION. ID: INSTANCE DISCRIMINATION. CL: CLUSTERING.

| | Method | Pretext Task | Code/PT |
|---|---|---|---|
| Images | RotNet [33] | TP | Y/Y |
| | iGPT [52] | MP | Y/Y |
| | Colorization [53] | MP | Y/Y |
| | Inpainting [26] | MP | Y/Y |
| | MoCo [40] | ID | Y/Y |
| | SimCLR [5] | ID | Y/Y |
| | BYOL [6] | ID | Y/Y |
| | SwAV [7] | Cl | Y/Y |
| Video/MM | VCP [54] | MP | Y/N |
| | CLIP [43] | ID | Y/Y |
| | XDC [9] | Cl | N/Y |
| | ViLBERT [55] | MP + ID | Y/Y |
| Text | word2vec [22] | MP | Y/Y |
| | ELMo [56] | MP | Y/Y |
| | BERT [12] | MP | Y/Y |
| | GPT [13], [32] | MP | Y/Y N/N |
| S & TS | CPC [11] | MP | N/N |
| | wav2vec [10] | MP | Y/Y |
| | STRN [34] | TP | Y/Y |
| Graph | Node2Vec [14] | MP | Y/N |
| | GraphSAGE [57] | MP | Y/N |
| | DGI [15] | ID | Y/N |
| | GPT-GNN [27] | MP | Y/Y |
| | GraphTER [24] | TP | Y/Y |

SSRL researchers, and to the broader AI community that make use of pre-trained features.

## V. METHODS AND DATASETS

In this section we review major methods and considerations broken down by data modality. Summaries of major methods and datasets for Image, Video, Text, Time-series and Graph modalities are provided in Table I and II respectively.

### A. Images

Computer vision tasks performed on still images vary broadly from recognition (whole image classification), detection (object localisaton within an image), and dense prediction (e.g., pixel-wise segmentation). State of the art performance on all of these tasks is achieved by supervised deep learning, and thus SSRL aims to alleviate the annotation bottleneck in computer vision by providing self-supervised pre-training that can be combined with data-efficient fine-tuning.

Computer vision has long been dominated by the use of convolutional neural networks (CNNs) that use weight-sharing to reduce the number of learnable parameters by exploiting the spatial properties of images. State of the art architectures usually start with CNN representation encoding $h_\theta(\cdot)$, with ResNet [1] being widely used, before appending task-specific decoding heads $g_\phi$. Many of the initially successful methods in self-supervised representation learning used ResNet backbones [5] but a recent trend has brought Transformer architectures into the vision domain [52]. One notable version is the Vision Transformer (ViT) [58] that is increasingly being used by recent self-supervised methods on image data [43].

TABLE II
COMMON SOURCE DATASETS USED IN EACH MODALITY.

| | Source | Size |
|---|---|---|
| Images | ImageNet [35] | 1.3M images |
| | YFC100M [59] | 100M images |
| | iNaturalist [60] | 2.7M images |
| Video/MM | Kinetics [61] | 650k videos |
| | YouTube8M [62] | 8M videos |
| | HowTo100M [63] | 136M videos |
| Text | WikiText [64] | 100M tokens |
| | OpenWebText [65] | 40GB of text |
| | Common Crawl [66] | 410B tokens |
| S/TS | LibriSpeech [67] | 960 hours of speech |
| | Libri-light [68] | 60K hours of speech |
| | AudioSet [69] | 5.8k hours of audio |
| Graph | Open Academic Graph [70] | 178M nodes, 2B edges |
| | Amazon Review Recommendation [71] | 113M nodes |
| | PROTEINS [72] | 1.1k graphs |

*1) Methods:* All types of pretext tasks (Section III) have been widely applied in still imagery (Table I). The earliest example of a self-supervised system, given the modern interpretation of the phrase, is the work of [37]. This paper introduced two fundamental ideas still relevant to techniques being developed today: (i) metric learning with a contrastive loss and a heuristic for generating training pairs that can be used to train a neural network feature extractor; (ii) using side-information, such as the relative position or viewing angle of training images, can be used to learn invariant or equivariant features. Subsequent methods that focused on SSRL for single images also pursued the goal of developing feature extractors that are invariant to different types of transformations, through transformation augmentations [23].

Several methods fall into the transformation prediction family, focusing on modifying unlabelled images using a known transformation, like rotation [33], and then training the network to predict the angle of that rotation. Others mask out information in the training images and require the network to reconstruct it, leading to pretext tasks such as colourisation [53] and inpainting [26], where colour channels and image patches are removed, respectively. A state of the art example in this category is iGPT [52], which exploits a self-attention architecture and masked prediction for representation learning.

The majority of recent methods focus on the relations between different images in the dataset, using instance discrimination [5], [40] or clustering [7], and heavy data augmentation has become a vital component required by all methods to achieve high performance. Progress has accelerated rapidly in the last two years, with the latest methods now systematically outperforming supervised pre-training in diverse downstream tasks and datasets [29], as shown in Figure 5.

*2) Datasets:* As in much of computer vision, ImageNet [35] is the most common source dataset for self-supervised pre-training [40], [5], [7], [6], consisting of 1.28 million training images across 1,000 object categories, with the most commonly used resolution at $224 \times 224$. Many methods are increasingly using datasets much larger than ImageNet. For example YFCC100M [59], with 100 million images from

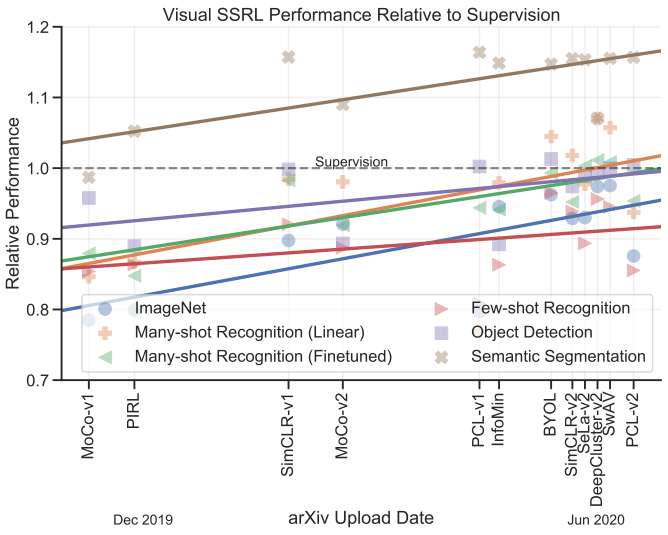


Fig. 5. The relative performance of SSRL methods on visual tasks, compared to a supervised baseline. Figure produced based on results in [29].

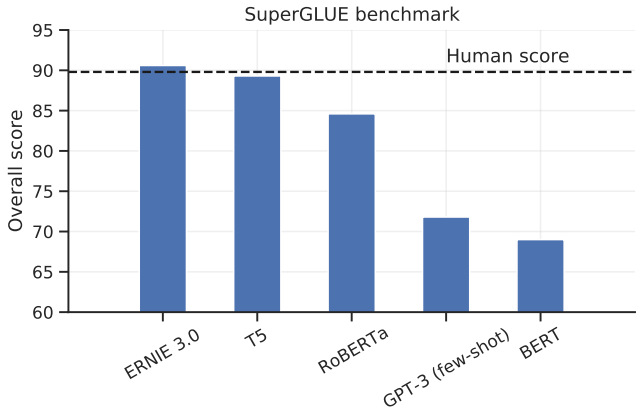


Fig. 6. The performance of SSRL methods on the textual benchmark SuperGLUE, compared to a baseline of human performance. Selected methods taken from the official leaderboard at https://super.gluebenchmark.com/leaderboard.

Flickr, used by [46], and [36] with 3.5 billion images from Instagram. Subsets of the latter are used by [40], [7].

The ImageNet benchmark is a highly curated dataset, with certain biases that do not appear in natural images, such as centring of objects and clear isolation of object from background. iNaturalist [60] is a collection of wildlife datasets compiled by a citizen science project, where members upload their own photographs and others collectively annotate them. This forms a more natural dataset which exhibits class imbalance and distractor objects which often complicate real-world tasks. While it has not yet served as the source dataset for any new method, it has been used to benchmark the robustness of existing SSRL methods to more uncurated data [73].

*3) Applications:* On established benchmarks, SSRL has had widespread and significant success in matching and surpassing supervised pre-training performance, especially for image recognition tasks, and in photo imagery of similar character to ImageNet (Figure 5). Progress in transfer to more diverse downstream tasks such as detection and segmentation; as well as downstream datasets which are out of distribution with respect to pre-training data has also been steady [29], if less rapid.

Beyond common benchmarks, SSRL has been successfully applied in application areas where labelled data is sparse, such as earth observation remote sensing [74]. In these cases, pre-training on the available unlabelled target-domain data was beneficial to compensate for sparse annotations. A growing downstream consumer of SSRL is the medical imaging domain, where labelled data is often intrinsically sparse or too expensive to collect in bulk for end-to-end learning from scratch. For example [75] used unlabelled brain scan images to perform image restoration (an inpainting-like task), improving upon random initialisation for fine-tuning several downstream tasks. A somewhat unique feature of the medical imaging domain is the processing of 3D volumetric images such as from MRI images. This has also recently inspired various extensions of standard pretext tasks into 3D [76].

A final application where SSRL pre-training has been successfully applied is that of anomaly detection. SSRL-based approaches typically either train a feature to be used in conjunction with a classic generative anomaly detector, or more interestingly use the SSRL objective itself to produce an anomaly detection score. For example, current state of the art anomaly detectors [77] rely on SSRL training of rotation prediction, with the rotation prediction accuracy providing the anomaly score.

### B. Video and Multi-modal

In the domain of video and multi-modality diverse tasks are of interest including video recognition, action/event detection (localisation of an event within a longer video), tracking (localising an object within frames and across time), and cross-modal retrieval (e.g., retrieving a video frame given associated subtitles). State of the art architectures again dominate all of these tasks given access to sufficient training data to train encoder and task-specific decoder components.

Common architectures $h_\theta(\cdot)$ for encoding videos include 3D CNNs or multi-stream encoders that process appearance and motion separately. In the case of multi-modal processing of video and audio, or video and associated text, one requires a synchronised video CNN encoder as well as a text/audio encoder (e.g., recurrent neural network) to encode the multi-modal streams. These data streams may then be fused into a single representation and decoded at each time-step (e.g., for localisation/detection), or first pooled over time (e.g., for video-level recognition).

*1) Methods:* Transformation prediction (TP) and contrastive instance discrimination methods are the most widely used for SSRL in video. There are a wide variety of TP pretexts in video. Rotation, and colorization discussed earlier are also widely generalised to video data. Making more unique use of the temporal nature of video, one can for example predict the ordering of frames or clips [8], or the speeding up or slowing down of videos.

In terms of contrastive instance discrimination methods, data augmentation has been the main mode of obtaining different views in still-imagery. However, for videos several methods exploit multiple sensory views, like RGB, optical flow, depth, and surface normals [41], [78] that provide different views for learning cross-view video clip matching.

A recent notable method in the instance discrimination family is CLIP [43], a visuo-linguistic multi-modal learning algorithm that has further advanced state of the art in robust visual representation learning by crawling pairs of images and associated text from the internet, and exploiting them for cross-view contrastive learning. Massive multi-modal pre-training was shown to lead to excellent performance on diverse downstream tasks including language-based image retrieval.

Clustering has been used in similar ways to match inputs from different modalities to the same clusters [9]. Finally, masked prediction has been applied through filling in masked out clips [54].

*2) Datasets:* There are several datasets of videos used for pre-training in this modality. Kinetics [61] is a large action recognition dataset of human-object and human-human interactions, collected from YouTube videos. One version, Kinetics-400, contains around 300k videos. There are larger versions of the dataset with up to 700 classes and 650k videos. Recently, a group of very large-scale datasets have been constructed from publically available videos on social platforms, like YouTube8M [62] and HowTo100M [63], the latter containing 136 million YouTube instructional videos with narration with captions across 23k visual tasks.

For methods using multiple modalities, the visual and audio information often come from the large video datasets discussed above [79]. An additional dataset that has been considered here is AudioSet [69], an audio event detection dataset. For methods using text information, this often obtained from automated transcription using ASR. Other datasets have textual information built in, such as subtitles or video descriptions.

*3) Applications:* As outlined in the previous section, the most common application and benchmark scenario for video SSRL is in video action recognition and detection in various guises. SSRL has made rapid progress in this area and state of the art methods trained on massive pre-training sources lead to significantly better performance than direct training on an array of standard benchmarks [9], but do not yet reliably surpass supervised pre-training on the same source datasets as in the case of still images earlier.

Similarly to the still image domain SSRL has been successfully applied to video anomaly detection. For example, given a TP pretext task of arrow of time prediction (differentiating forward vs reverse frame sequences) among others, videos with high probability of being reversed can be considered anomalous [17].

Video data is often multi-modal, covering RGB+D, video+audio, or video+text (e.g., from script or text2speech) modalities. It is noteworthy that several studies [41], [79] have explored how SSRL on multi-modal source data can be used to learn a stronger representation for single-modality downstream tasks, and ultimately outperform single-modality pre-training on diverse downstream tasks in uni-modal video, still-image, or audio domains [79].

With regards to the video and text, several recent SSRL studies have learned joint multi-modal representations. Notably, ViLBERT [55], exploited both BERT-like masked prediction and contrastive instance discrimination to learn a multi-modal representation which then achieved state of the art performance in downstream vision and language tasks such as caption-based retrieval, visual question answering, and visual commonsense reasoning.

### C. Text and Natural Language

Natural Language Processing (NLP) methods aim to learn from raw input text and solve a wide variety of tasks ranging from low-level such as word-similarity, part of speech tagging and sentiment; to high-level tasks such as question answering and language translation. State of the art approaches are often based on deep sequence encoders such as LSTM, and in recent years self-attention based approaches have dominated [2]. With data annotation being a major bottleneck, NLP was the first disciplines to make major and successful use of self-supervision [22].

*1) Methods:* Self-supervised representation learning has been a fundamental component in natural language processing (NLP) for many years. Masked prediction methods have been particularly effective in this modality, with word embeddings becoming widely adopted as they succeed in producing representations that capture the semantic similarity of words, as well as being able to deal with arbitrary vocabulary sizes. word2vec [22] and related methods work by either predicting a central word given its neighbours – called *continuous bag of words* (CBOW) – or predicting the neighbours given the central word – called *skip-gram*. Given such pre-trained word embeddings, a target task is then solved by mapping input tokens to their vector embeddings and learning a model on top of them. Since the embedding for a word is fixed after training, it cannot adapt to the context in which the word appears, causing a problem for words with many meanings.

As opposed to these non-contextual embedding methods, topical contextual methods learn embeddings which change depending on the surrounding words. The two most common approaches to this are next word prediction [13] and masked word prediction [12], with the landmark BERT method combining the latter with next sentence prediction [12]. For the encoder architecture, recurrent networks like LSTMs [18] have long been used to model the context while recent works have moved to Transformer-based architectures with self-attention [2], which allow longer range connections to be made across words in a sentence, but require more data for training. A final trend is that new models are becoming bigger and bigger, counting ELMo [56] at 94M, BERT [12] at 340M, GPT-2 [13] at 1.5B and GPT-3 [32] at 175B parameters. Recent progress on this type of large-scale masked prediction has led to performance surpassing human baselines on language understanding tasks. This can be seen in Figure 6 where we show the performance of selected top models from the leaderboard of the common SuperGLUE [80] benchmark.

All methods discussed above belong to the masked prediction family of methods and they have been the most successful and widely adopted. But there are examples of transformation prediction such as recovering the order of permuted [81] or rotated [81] sentences. These have often been used as complementary signals in order to improve downstream performance on a particular task.

*2) Datasets:* Self-supervision in language has shown to benefit from ever larger corpora of text. This has led to huge datasets being created, primarily by crawling the web for the data. Early word embeddings made heavy use of Wikipedia articles [64], or crawls of news sites and social media sites like Twitter. As models have become larger and require more text to train on, the organisations training these models have begun using private datasets which are not publically available [13], [32]. Attempts have been made at replicating the data used in such papers, for example the OpenWebText Corpus [65]. Another example is Common Crawl [66], a non-profit project that makes data from billions of web pages freely accessible. Various datasets have been created from this data and filtered versions of the entire corpus often form the bulk of training sets [32]. Using a combination of the above data sources, the total size of the training set used in state-of-the-art language modeling is now on the scale of 500 billion tokens [32].

*3) Applications:* SSRL has made a major impact on a host of problems involving multiple languages, which introduces a new kind of source/target dichotomy besides the task and domain-level dichotomies we have focused on thus far. In the simplest (within-language) scenario, SSRL can benefit all the standard language understanding tasks (classification, QA, etc) for *low resource languages*. One can pre-train SSRL models on a large corpora of high-resource languages before fine-tuning them on smaller corpora of low-resource languages [25]. For cross-language tasks such as machine translation, one can pre-train SSRL models (e.g., for masked prediction) that are multi-lingual, in that they simultaneously encode/decode data from more than one language. These multi-lingual language models are then well primed for comparatively low-data fine-tuning for translation [25], or provide good representations to drive unsupervised [25] learning of translation models. This is valuable, as vanilla translation models are extremely expensive to supervise due to requiring a vast number of aligned (translated) sentence pairs across languages.

The conventional task-specific fine-tuning as outlined in Section II is the dominant paradigm for exploiting SSRL in language. However, A notable exception to this is in the recent GPT-3 [32] language model. A key observation in this work is that a sufficiently scaled-up 175B parameter generative language model can often perform few or zero-shot learning of a new task in a purely feed-forward manner (no back-propagation or fine-tuning) simply by prompting the model with the few-training examples, and the query and allowing it to complete the answer.

*4) Considerations:* A growing concern in language modeling is the extent to which biases implicit in the large training corpora for SSRL become baked into the resulting language models, for example sexist or racist stereotypes. Vast corpora must be used for SSRL, so training data cannot be filtered for appropriateness manually. A small but growing body of work aims to develop SSRL variants with reduced bias [82].

### *D. Audio & Time-series*

Classic approaches to audio analysis tasks such as speech recognition compute mel-frequency cepstrum coefficients (MFCC) from the raw audio data and then models the sequence via Gaussian Mixture Models and Hidden Markov Models. Meanwhile contemporary neural network approaches trained by supervised learning have dominated in settings where massive annotated training data is available [83]. Against this backdrop, self-supervised methods have very recently made massive advances in alleviating this annotation bottleneck, enabling state of the art audio analysis methods to be trained with relatively sparse annotations.

Self-supervised methods in the audio analysis arena have exploited architectures $h_\theta$ spanning all the popular options for time-series data including recurrent [84], convolutional [11] and self-attention [10] networks. These are usually applied directly to raw waveform data to build a representation without any pre-processing step such as MFCC.

*1) Methods:* In terms of self supervision algorithms, numerous studies have successfully adapted the insights of self-attention based language models [12] to audio data. As a pretext task, random segments of the input sequence are masked and predicted by a self-attention architecture. However, a key difference is that language models work with discrete token sequences – thus enabling the pretext to be formalised as multi-class classification, while audio time-series are naturally continuous. Thus solutions to formalising a masked prediction task for audio have either quantised the speech embedding for classification – such as Wav2Vec-2.0 [10]; applied contrastive losses to differentiate the masked segment from alternative distractors – such as CPC [11]; or replaced classification-based prediction with regression layers to directly synthesise the masked frame – such as APC [84]. Other approaches such as PASE [85] go beyond defining a single self-supervision pretext task to combines several losses, each predicting or classifying a different property of the input.

*2) Datasets:* While it is not as strong as in the text modality, there is still a trend for newer models to train on larger and larger datasets. Small datasets historically used for model model training, are now reserved for downstream evaluation, with contemporary mehods being pre-trained on LibriSpeech [67] containing 960 hours of speech from audiobook readings and Libri-light [68], a much larger dataset (60K hours) of similar audiobook recordings.

*3) Applications:* A notable success in Speech was shown by Wav2Vec 2.0 [10] which used transformers+masked prediction SSRL on 53k hours of unlabelled data prior to fine-tuning a downstream speech recognition system. This was subsequently able to surpass prior state of the art ASR performance with 10-fold less supervised data than used before, and approach state of the art with 100-fold less supervised data than before. Albeit at the cost of 660 GPU-days of SSRL compute, this is a dramatic improvement in data efficiency.

In terms of more general time-series data representation learning, masked prediction methods based on transformer

architecture have been shown to match supervised state of the art in a variety of diverse benchmarks a suite of benchmarks in diverse application areas [86].

A major application area for self-supervised time-series analysis is medical data, where annotations are hard to collect. There has been progress in applying SSRL to EEG and ECG data [34], [87]. For example using transformation prediction SSRL prior to training ECG-based emotion recognition [34] and contrastive instance discrimination SSRL prior to learning downstream EEG-based motor movement classification and ECG-based anomaly detection [87]. In terms of time-series forecasting, transformers based on sequential masked prediction pretext have significantly outperformed conventional autoregressive models in predicting disease transmission [88].

### E. Graphs

Graph structured data is ubiquitous in the networked world, and supports a diverse array of tasks including node, edge, and graph classification. These tasks should all be informed by both node/edge features where available, and graph connectivity. Graph Neural Networks [89] have advanced all these tasks significantly, especially where massive labelled data is available. Thus a large body of work on self-supervised graph representation learning has emerged to facilitate downstream GNN-based tasks.

Graph-based SSRL can be somewhat unique in several aspects. Depending on whether the ultimate task of interest requires node-level or graph-level predictions, methods may focus on learning node-level [15], [57], [27] or graph-level [90] representations, or both [91]. Graph-based methods also differ in whether they are oriented at training on a *set* of graphs [15], [90] (cf: set of images or audio clips in other modalities), or on a single large graph [14].

*1) Methods:* Early shallow methods for self-supervised graph representation learning used NLP-inspired masked prediction approaches to learn node-embeddings, for example based on random walks on the graph [14]. Much as shallow word embeddings have been eclipsed by deep language models in NLP, newer graph-representation learning architectures that focus on graph convolutional networks or self-attention have driven progress in this modality.

In terms of self-supervision objectives, most work in this area falls into masked prediction and instance discrimination categories. Several recent methods optimise mutual information-based instance discrimination objectives, with DGI [15] and InfoGraph [90] performing contrastive instance discrimination between pairs of nodes/patches and whole graphs. Masked prediction pretexts were used both by classic shallow methods [14], as well as recent deep approaches such as GPT-GNN [27]. A minority of approaches have applied transformation-prediction methods – such as GraphTER [24], where node-wise transformations are applied and and predicted by a GNN.

An important dichotomy in graph-based representation learning is between transductive and inductive graph representation learning methods. The majority of methods are transductive, in that they learn embeddings specifically for nodes seen during, and so are primarily relevant in applications where the downstream task uses the same graph data as is used for pre-training. This is analogous to how the word2vec algorithm [22] in language learns embeddings for words in its training set, but cannot produce embeddings for unseen words. A minority of methods are inductive [57], [27] in that they learn embedding functions that do not depend on a specific choice of input graph, and thus can be transferred to new target nodes or graphs.

*2) Datasets:* Since the graph structured data occurs so pervasively, it covers a wide range of data types tasks. Major examples include social [57], citation [70], chemical [92] and biological networks [93]. Because there are many different kinds of graphs with different structures and sizes there is no one-size-fits-all source dataset which consistently improves transfer as in many of the previously discussed modalities. It instead depends on the tasks of interest.

For learning in the transductive setting, the pre-training must necessarily be done on the same graph as the testing, thus limiting the transfer task-transfer and not domain-transfer. For the inductive setting, the source data can differ from target but in most evaluation cases the test set consists of nodes that were hidden from the training graph [27] or unseen graphs from the same underlying dataset [93]. Like in other modalities we have seen increasingly large graphs being used for pre-training, like the Amazon Review Recommendation data [71] with 113 million nodes or Open Academic Graph (OAG) [70] consists of over 178 million nodes and 2 billion edges.

*3) Applications:* Self-supervised graph-based representation learning is expected benefit all graph-based prediction applications where data is limited. This is especially the case in computational chemistry and biology applications, where graphs and associated annotations may correspond to molecules and corresponding molecular properties. In such applications data are intrinsically hard to collect, but predicting graph properties can significantly impact tasks such as drug discovery and material discovery [92], [93]. In computer vision, using LIDAR rather than RGB sensors leads to observations represented as point clouds or graphs rather than conventional images. In this case self-supervised graph representation learners such as GraphTER [24] have led to excellent performance in object segmentation (i.e., node classification) and classification (i.e., graph classification).

## VI. Discussion

### A. Pre-training Cost

The pre-training cost of different SSRL methods is not consistently documented, and hardware platform/GPU differences make them hard to compare quantitatively. Nevertheless, it is clear that we can see that state of the art methods in computer vision, speech and text (Tables I and II) require massive resources on the order of 100s of GPU-days for training on ImageNet, LibriSpeech, and Wikipedia corpora respectively. The general purpose pre-trained nature of these representations may amortise this cost somewhat, by enabling many downstream problems to be solved with the same representation. This has largely been the case in the text

modality where there has been strong success fine-tuning generic pre-trained models to diverse tasks [12]. However this may not be possible in other modalities such as graphs which may require transductive training, or vision where domain-specific pre-training may be necessary for data very different to ImageNet such as hyperspectral imagery or volumetric MRI. In this case pre-training cost poses an accessibility barrier to modestly resourced organisations, and an environmental issue [94] due to its energy requirement. While there is also tremendous research activity in developing more efficient pre-training algorithms, the net cost of pre-training is trending upward due to the fact that bigger datasets and bigger network architectures have systematically led to better performance.

### B. Data Requirement and Curation

For text [32] and speech [10] the literature unambiguously shows that thus far performance increases consistently with ever larger datasets. In the case of text, this result further seems to be relatively insensitive to the degree of curation of the data.

For images, the majority of recent work still uses still uses ImageNet with its 1.28 million images as the source set [7], [6]. However, a number of studies have shown that using larger pre-training datasets [59], [36] benefits to transfer performance [36], [16], with feature quality growing logarithmically with data volume [16]. For video pre-training, the state-of-the-art models use the increasingly large YouTube8M-2 [62] and HowTo100M [63] with combined video play-times of 13 and 15 years respectively.

The vision of SSRL is to enable representation learning on easily obtained uncurated data. However, for benchmarking purposes (especially in vision and audio and graphs, but less so in text), methods are often actually trained on curated data while ignoring the labels. It is not clear how much existing algorithm design is overfitted to these curated datasets, and if the relative performance of different methods is maintained when real uncurated data is used instead. For example in computer vision, most pre-training is performed on ImageNet, which is large and diverse, yet uniformly focused on individual objects. If this was replaced with scene images with multiple cluttered objects, then typical instance discrimination tasks like mapping two different crops of one image to the same identity could create false positive pretext label noise that maps different semantic objects to the same representation [95]. We are beginning to see new SSRL methods designed for data with different statistics such as cluttered images [95].

### C. Architecture Choice and Deployment Costs

For both image [5], [16] and text [12] analysis, the trend has been that bigger architectures lead to better representation performance, especially when coupled with extremely large pre-training datasets, and challenging pretext tasks [16]. This is welcome from the perspective of near 'automatic' performance improvement as datasets and compute capabilities grow. However, it does pose a concern for deployment of the resulting models on resource-constrained or embedded devices with limited memory and/or compute capability, which may limit the benefit of this line of improvement for such applications.

A standard approach to alleviate this issue is to perform SSRL of large models as usual followed by using unlabelled data to perform post-training *distillation* of the large self-supervised model into a smaller more compact but similarly performant student model. For example, in vision this has been demonstrated to compress a ResNet-152×3 model to a ResNet-50 of similar performance [5]; in text a 109M parameter/22.5GFLOP BERT model can be distilled to a 14.5M/1.2GFLOP BERT model with similar performance [31].

### D. Transferability

The vision of SSRL is to produce features that transfer to a wide range of downstream tasks. The extent to which this has been realised varies by discipline/modality. In vision this is on its way with many studies evaluating transfer performance [29], [16], but no single benchmark has yet been widely agreed. Recognition has been the most common scene of transfer assessment but recently detection and dense prediction have also been embraced [7], [6], [29]. However, ImageNet Top-1 accuracy is still the main metric used in model comparisons. As reported by [29], this metric shows high correlation with downstream recognition performance. Their results for detection and dense prediction, however, show markedly lower correlations, indicating that current SSRL methods are not optimised for such a broad transfer [29]. For practitioners with new data and tasks, this means that the best performing SSRL model on ImageNet can safely be adapted to recognition tasks. However, if the task differs then more models need to be considered. Additionally, if the images of the target domain are unstructured or exhibit different properties to ImageNet images, then further caution must be taken when choosing a pre-trained model. This is further expanded on by [73] who show how SSRL models fail to compete with supervision on in-the-wild datasets containing plant and animal species, contrasting what has been found for curated datasets [29].

In video and multi-modal settings, common transfer evaluation considers transfer from large source datasets such as Kinetics [61] to standard target datasets such as UCF101. State of the art methods successfully leverage large source datasets and approach but do not yet outperform supervised pre-training [78]. Nonetheless, there has been an uptake of SSRL methods in applications such as tracking [96] and detection.

In text, the field has matured more already. Here, several broad benchmarks such as SuperGLUE [80] are regularly used to monitor progress. The main mode of transfer in NLP has long been to fit a linear model or fine-tune a SSRL model like BERT [12], and on many tasks on the above benchmarks fine-tuned SSRL models achieve top results.The recent GPT-3 [32] has shown that huge SSRL models can achieve competitive performance via few-shot adaptation instead of full fine-tuning, especially on language modeling and question answering. In summary, text models exhibit relatively high transferability with SSRL pre-training dominating in a broad range of downstream tasks.

In speech and time-series the focus has so far been narrow, with only a few tasks and datasets forming the evaluation landscape. These cover phoneme recognition and occasionally speaker identification or emotion classification. Most work focuses on the English language speech both for pre-training and transfer. However very rapid progress is currently being made in multi-lingual [97] speech models and cross-lingual transfer [98], so prospects for transferability seem promising.

The current state of the graph modality is that transferability is good to unseen nodes within the same graph and to unseen graphs within the same dataset, e.g. PPI [93]. However, there is little information to suggest transfer across graph types, like chemical-to-biological or citation-to-social, currently has any benefit.

### *E. Choosing the Right Pretext Task*

As we have seen, the four families of pretext tasks can be applied to all the different modalities. But because self-supervised pretexts rely on exploiting the structure of data, which in turn differs significantly across modalities, their efficacy can vary substantially across modalities. One clear such trend is that masked prediction is ubiquitous in the text modality [22], [12], [32], with other tasks being significantly less effective. And when other tasks are used, they are often complementary to a masked prediction loss [81]. In images, masked prediction and transformation prediction have been tried in various forms and drove initial progress, but the most recent advances in these modalities have been driven by instance discrimination [5], [78] and clustering [7], [9]. However, transformation prediction is still seeing success in videos, presumably because of of the rich spatio-temporal information to be exploited. Finally, while there may be a dominant pretext strategy for a given modality, it is common that a suitably designed combinations of pretexts applied in a multi-task manner can improve performance compared to a single pretext [81].

Picking a pretext based on the bulk of successes for the modality of interest is a good start. However to further inform choice, one can further consider the assumptions which underlie each family of methods. Masked prediction relies on context being enough to fill in missing parts of a datapoint. Transformation prediction relies on each datapoint possessing a canonical view. Instance discrimination relies on each datapoint representing a unique semantic example, distinguished from all other datapoints in the training set, which may not hold for cluttered images as discussed above. It is notable that clustering requires no strong assumptions other than the existence of meaningful similarities by which to group the data into a certain number of clusters. Therefore, if little is known of the structure of the data, then a method based on clustering may be a good start.

A final consideration when selecting a pretext task is *what properties do we want in our representations?* If our data modality is images and we are interested in exploiting the orientation of objects in our data, do we want our representations to vary with orientation – in which case we might want to use a transformation prediction method like [33] – or do we want all orientations of the input to produce the same output – in which case we might instead choose an instance discrimination method that uses rotation-based augmentation. This question of *equivariance* or *invariance* can greatly impact the downstream performance of certain tasks. For example, a visual object classification task might benefit from invariance to spatial translation but a detection task would need this information to be preserved in order to correctly predict object locations.

If there are no specific downstream task in mind a-priori and therefore no known required properties that must be learned, the ideal selection is not clear. In this case we want to use the method which best captures the core information in our data which has the best chance of being of use for later tasks. Finding such pretext tasks can be considered the main aim of the self-supervised representation learning field of research.

### *F. Self-supervised vs Semi-supervised*

In cases where the source and target datasets are the same or similar in content and label-space, then both *semi-supervised* and *self-supervised* approaches can potentially apply (Section II). Since both families of methods are making rapid progress and there have been few direct comparisons, it is not yet clear if/when one family should be preferred. However, since SSRL deals with initialisation and SSL deals with refinement, the two strategies can in principle both be applied to one learning problem. There has yet been very little investigation into the extent to which these strategies can be complementary and further boost performance when used together. A preliminary result in computer vision suggests not [99]. However, preliminary results in text [100] suggest that SSL and SSRL can be synergistic when used together.

### *G. Other Benefits of SSRL*

While we have mainly focused on the benefits of SSRL with respect to accuracy in the low and few-shot data regime, there are several other potential benefits: (i) The computational cost of fine-tuning a self-supervised model tends to be lower than training from scratch (though comparable to fine-tuning a supervised pre-trained feature). (ii) If the supervised target task suffers from label-noise, training leads to much worse performance compared to using clean labels. However SSRL also increases resilience to such label noise [28], which often occurs in practice. (iii) Given a trained system, SSRL can also improve robustness of image recognition to adversarial attack, as well as common corruptions such as blur, noise, and compression artefacts [28]. (iv) Furthermore SSRL leads to better calibrated probabilities [29], [28], which can be used to drive abstention of automated predictions, or out-of-distribution detection [28]. (v) Finally, in terms of model interpretability feature extractors trained by self-supervision tend to lead to more reasonable and interpretable attention maps [29].

### H. Recommendations for Future Work

- Develop wider benchmarks. Several of the modalities we look at have a few standard downstream tasks that are consistently evaluated against. This creates a bias towards making new methods that optimise only for those particular tasks. Instead we should create benchmark suites that study the performance of pre-trained models across a wide range of tasks within a modality. This has been successfully done in NLP and has driven progress and made sure it benefits many areas of the field [80], but such standardised benchmarks are lacking in the other modalities we have considered.
- Do not only focus on tracking task performances in these benchmarks but also track other feature properties like social biases to obtain broader understanding of how these models behave. Progress on reducing such biases can only really be done if we know about and can quantify them.
- Be wary of relying only on scale to improve performance. As we use larger and larger datasets to train these models, we know less and less about the data itself as there is very little human oversight in the data collection process. By developing methods that are more data efficient – i.e. don't need billions of instances to learn – we can create models that are easier to understand and control. Additionally, as we develop larger models their carbon footprint grows significantly [94]. Make sure that the efficiency of training these models is tracked in common benchmarks.
- Do not get stuck on training on only one specific source dataset as this will bias the type of methods that are created. As an example, the highly curated and single-centred-object-style of ImageNet has led to a particular style of data augmentation and instance discrimination. However, it has been shown that on less curated 'in-the-wild' images, these methods underperform. By continuously considering different types of source datasets we get a better picture of when and where a method works.

## Acknowledgements

This research was partially supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1 and the MOD University Defence Research Collaboration (UDRC) in Signal Processing; EPSRC Centre for Doctoral Training in Data Science, funded by EPSRC (grant EP/L016427/1) and the University of Edinburgh; and EPSRC grant EP/R026173/1.

## References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.
[3] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *ICCV*, 2017.
[4] L. Jing and Y. Tian, "Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
[5] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big Self-Supervised Models are Strong Semi-Supervised Learners," in *NeurIPS*, 2020.
[6] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning," in *NeurIPS*, 2020.
[7] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised Learning of Visual Features by Contrasting Cluster Assignments," in *NeurIPS*, 2020.
[8] D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang, "Self-supervised Spatiotemporal Learning via Video Clip Order Prediction," in *CVPR*, 2019.
[9] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, "Self-Supervised Learning by Cross-Modal Audio-Video Clustering," in *NeurIPS*, 2020.
[10] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *NeurIPS*, 2020.
[11] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," *arXiv*, 2018.
[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *NAACL*, 2019.
[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," Tech. Rep., 2019.
[14] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *KDD*, 2016.
[15] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengioy, and R. D. Hjelm, "Deep Graph Infomax," in *ICLR*, 2019.
[16] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and Benchmarking Self-Supervised Visual Representation Learning," in *ICCV*, 2019.
[17] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "Anomaly detection in video via self-supervised and multi-task learning," *CVPR*, 2021.
[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 2016.
[19] H. Gouk, T. M. Hospedales, and M. Pontil, "Distance-based regularisation of deep networks for fine-tuning," in *ICLR*, 2021.
[20] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, 2020.
[21] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2009.
[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *NeurIPS*, 2013.
[23] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks," in *NeurIPS*, 2014.
[24] X. Gao, W. Hu, and G.-J. Qi, "GraphTER: Unsupervised Learning of Graph Transformation Equivariant Representations via Auto-Encoding Node-wise Transformations," in *CVPR*, 2020.
[25] A. Conneau and G. Lample, "Cross-lingual language model pretraining," in *NeurIPS*, 2019.
[26] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," in *CVPR*, 2016.
[27] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, "GPT-GNN: Generative Pre-Training of Graph Neural Networks," in *KDD*, 2020.
[28] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *NeurIPS*, 2019.
[29] L. Ericsson, H. Gouk, and T. M. Hospedales, "How Well Do Self-Supervised Models Transfer?" in *CVPR*, 2021.
[30] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting Self-Supervised Visual Representation Learning," in *CVPR*, 2019.
[31] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for Natural Language Understanding," in *EMNLP*, 2020.
[32] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," in *NeurIPS*, 2020.

[33] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations," *ICLR*, 2018.

[34] P. Sarkar and A. Etemad, "Self-supervised Learning for ECG-based Emotion Recognition," in *ICASSP*, 2020.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[36] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, "Exploring the Limits of Weakly Supervised Pretraining," in *ECCV*, 2018.

[37] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *CVPR*, 2006.

[38] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," *Journal of Machine Learning Research*, 2010.

[39] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow Twins: Self-Supervised Learning via Redundancy Reduction," in *ICML*, 2021.

[40] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *CVPR*, 2019.

[41] Y. Tian, D. Krishnan, and P. Isola, "Contrastive Multiview Coding," in *ECCV*, 2020.

[42] A. Owens and A. A. Efros, "Audio-Visual Scene Analysis with Self-Supervised Multisensory Features," in *ECCV*, 2018.

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

[44] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning," in *NeurIPS*, 2020.

[45] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," in *NeurIPS*, 2020.

[46] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep Clustering for Unsupervised Learning of Visual Features," in *ECCV*, 2018.

[47] X. Zhan, J. Xie, Z. Liu, Y. S. Ong, and C. C. Loy, "Online Deep Clustering for Unsupervised Representation Learning," in *CVPR*, 2020.

[48] N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar, "A theoretical analysis of contrastive unsupervised representation learning," in *ICML*, 2019.

[49] N. Saunshi, S. Malladi, and S. Arora, "A mathematical exploration of why language models help solve downstream tasks," in *ICLR*, 2021.

[50] J. D. Lee, Q. Lei, N. Saunshi, and J. Zhuo, "Predicting what you already know helps: Provable self-supervised learning," *arXiv preprint arXiv:2008.01064*, 2020.

[51] U. von Luxburg, "Clustering stability: An overview," *Foundations and Trends in Machine Learning*, vol. 2, no. 3, pp. 235–274, 2010.

[52] M. Chen, A. Radford, J. Wu, H. Jun, P. Dhariwal, D. Luan, and I. Sutskever, "Generative Pretraining From Pixels," in *ICML*, 2020.

[53] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *ECCV*, 2016.

[54] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang, "Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning," in *AAAI*, 2020.

[55] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *NeurIPS*, 2019.

[56] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL*, 2018.

[57] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *NeurIPS*, 2017.

[58] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *ICLR*, 2021.

[59] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "YFCC100M: The New Data in Multimedia Research," *Communications of the ACM*, 2015.

[60] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The iNaturalist Species Classification and Detection Dataset," in *CVPR*, 2018.

[61] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics Human Action Video Dataset," *arXiv*, 2017.

[62] "YouTube-8M: A Large and Diverse Labeled Video Dataset for Video Understanding Research." [Online]. Available: http://research.google.com/youtube8m/

[63] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips," in *ICCV*, 2019.

[64] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer Sentinel Mixture Models," in *ICLR*, 2017.

[65] A. Gokaslan and V. Cohen, "OpenWebText Corpus."

[66] C. Buck, K. Heafield, and B. v. Ooyen, "N-gram Counts and Language Models from the Common Crawl," in *LREC*, 2014.

[67] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[68] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-Light: A Benchmark for ASR with Limited or No Supervision," in *ICASSP*, 2019.

[69] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[70] F. Zhang, X. Liu, J. Tang, Y. Dong, P. Yao, J. Zhang, X. Gu, Y. Wang, B. Shao, R. Li, and K. Wang, "OAG: Toward Linking Large-scale Heterogeneous Entity Graphs," in *KDD*, 2019.

[71] J. Ni, J. Li, and J. McAuley, "Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects," in *EMNLP*, 2019.

[72] K. M. Borgwardt, C. S. Ong, S. Schonauer, S. V. N. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, 2005.

[73] O. Mac Aodha, "Benchmarking Representation Learning on Natural World Image Collections," in *CVPR*, 2021.

[74] C. Tao, J. Qi, W. Lu, H. Wang, and H. Li, "Remote sensing image scene classification with self-supervised paradi under limited labeled samples," 2020.

[75] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "Self-supervised learning for medical image analysis using image context restoration," *Medical image analysis*, 2019.

[76] A. Taleb, W. Loetzsch, N. Danz, J. Severin, T. Gaertner, B. Bergner, and C. Lippert, "3d self-supervised methods for medical imaging," *NeurIPS*, 2020.

[77] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *NeurIPS*, 2018.

[78] T. Han, W. Xie, and A. Zisserman, "Self-supervised Co-Training for Video Representation Learning," in *NeurIPS*, 2020.

[79] J.-B. Alayrac, A. Recasens, R. Schneider, R. Arandjelovi, J. Ramapuram, J. De Fauw, L. Smaira, S. Dieleman, and A. Zisserman, "Self-Supervised MultiModal Versatile Networks," in *NeurIPS*, 2020.

[80] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems," in *NeurIPS*, 2019.

[81] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *ACL*, 2020.

[82] P.-S. Huang, H. Zhang, R. Jiang, R. Stanforth, J. Welbl, J. Rae, V. Maini, D. Yogatama, and P. Kohli, "Reducing sentiment bias in language models via counterfactual evaluation," in *EMNLP*, 2019.

[83] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *ICML*, 2016.

[84] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An Unsupervised Autoregressive Model for Speech Representation Learning," in *Interspeech*, 2019.

[85] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning Problem-agnostic Speech Representations from Multiple Self-supervised Tasks," *Interspeech*, 2019.

[86] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," in *NeurIPS*, 2019.

[87] J. Y. Cheng, H. Goh, K. Dogrusoz, O. Tuzel, and E. A. Apple, "Subject-Aware Contrastive Learning for Biosignals," *arXiv*, 2020.

[88] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," in *ICML*, 2020.

[89] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *ICLR*, 2017.

[90] F.-Y. Sun, J. Hoffmann, V. Verma, and J. Tang, "InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization," in *ICLR*, 2020.

[91] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for Pre-training Graph Neural Networks," in *ICLR*, 2020.
[92] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: A benchmark for molecular machine learning," *Chemical Science*, 2018.
[93] M. Zitnik and J. Leskovec, "Predicting multicellular function through multi-layer tissue networks," *Bioinformatics*, 2017.
[94] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *Communications of the ACM*, 2020.
[95] S. Purushwalkam and A. Gupta, "Demystifying Contrastive Self-Supervised Learning: Invariances, Augmentations and Dataset Biases," in *NeurIPS*, 2020.
[96] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *CVPR*, 2019.
[97] A. Kannan, A. Datta, T. N. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, Z. Chen, and S. Lee, "Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model," *Interspeech*, 2019.
[98] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP*, 2020.
[99] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, "Rethinking pre-training and self-training," in *NeurIPS*, 2020.
[100] J. Du, E. Grave, B. Gunel, V. Chaudhary, O. Celebi, M. Auli, V. Stoyanov, and A. Conneau, "Self-training improves pre-training for natural language understanding," in *NACL*, 2021.