DABS: A Domain-Agnostic Benchmark for Self-Supervised Learning

Alex Tamkin†
Stanford University

Vincent Liu
Stanford University

Rongfei Lu
Stanford University

Daniel Fein
Stanford University

Colin Schultz
Stanford University

Noah Goodman
Stanford University

Abstract

Self-supervised learning algorithms, including BERT and SimCLR, have enabled significant strides in fields like natural language processing, computer vision, and speech processing. However, the domain-specificity of these algorithms means that solutions must be handcrafted for each new setting, including myriad healthcare, scientific, and multimodal domains. To catalyze progress towards more domain-agnostic methods, we introduce DABS: a Domain-Agnostic Benchmark for Self-supervised learning. To perform well on DABS, an algorithm must be pretrained on six unlabeled datasets from diverse domains: natural images, text, speech recordings, medical imaging, multichannel sensor data, and paired text and images, and then perform well on a set of labeled tasks in each domain. We also present e-Mix and ShED: two baseline domain-agnostic algorithms; their relatively modest performance demonstrates that significant progress is needed before self-supervised learning is an out-of-the-box solution for arbitrary domains. Code for benchmark datasets and baseline algorithms is available at [redacted].

Figure 1: The DABS Benchmark. A domain-agnostic self-supervised algorithm consists of 1) a model architecture, 2) an objective used to pretrain the model on unlabeled data, and 3) a transfer method used to deploy it on a downstream task (bolded items). A successful algorithm will achieve high performance on downstream tasks while holding these components constant across domains.

†atamkin@stanford.edu

1 Introduction and Motivation

Self-supervised learning (SSL) is on the rise across machine learning (ML), with notable recent successes in computer vision [16,36,33], natural language processing (NLP) [22,93,20], and speech processing [28,3]. SSL enables a model to acquire useful capabilities from unlabeled data; these capabilities can then be leveraged to drastically reduce the amount of labeled data needed to achieve high performance in a domain—a crucial advance given the time and expense needed to annotate datasets of millions of labels.

However, the potential impact of SSL is arguably greatest outside of the fields where it has currently seen the most success. Medical and scientific domains, for example, are rich in unlabeled data, yet the time and expertise needed for annotation far exceeds that for computer vision or NLP. This means that methods which reduce the need for labeled data are especially impactful in these settings.

Unfortunately, the most popular SSL methods are currently domain-specific—for example, the color jitter distortions used in SimCLR [16] are inapplicable to black-and-white chest x-rays, and the masked language modeling task used in BERT [22] is not directly applicable to spoken language, which is untokenized. Furthermore, these algorithms are challenging to develop, requiring costly trial-and-error by ML experts [16]. Unfortunately, while a great number of domains may benefit from SSL, this distribution exhibits a long tail where the vast majority of domains lack the ML expertise and resources to develop custom SSL solutions.

We argue that an appealing alternative to developing domain-specific SSL methods is to develop domain-agnostic techniques which work across a wide range of settings without extensive modification. Such domain-agnostic SSL algorithms could benefit the field in multiple impactful ways:

1. Making SSL work out-of-the box (Figure 2 right). The most important impact of domain-agnostic SSL would be turning SSL into an out-of-the-box technology capable of being used in any domain of interest without significant ML expertise. Aside from medical and scientific domains, this would also benefit the combinatorial number of multimodal settings which currently require novel algorithms to learn the relationships between modalities [55,17,26].

2. Improving handcrafted SSL methods (Figure 2 left). Several works have investigated how more general SSL methods can be combined with domain-specific knowledge (e.g. image augmentations) to provide gains [48,75,80]. This suggests that advances in domain-agnostic SSL could benefit popular ML domains as well, through combination with domain-specific methods.

3. Uncovering fundamental principles of SSL across domains. Communities such as computer vision and NLP currently have relatively disjoint investigations into SSL methods; this may obscure common scientific principles underlying the success of algorithms across modalities. Research on domain-agnostic methods may discover these general principles, which could benefit all domains.

However, despite the promise of domain-agnostic SSL, there has been no standardized way to evaluate or drive progress in a cross-cutting way among different communities. To fill this need, we propose DABS, a Domain Agnostic Benchmark for Self-supervised learning. DABS measures how well a single SSL algorithm works on many different domains, as opposed to just one. The benchmark is comprised of six domains representing different kinds of data: natural images, text, speech, medical imaging, multichannel sensor data, and captioned images. Each domain contains one unlabeled dataset for self-supervised learning, and at least one labeled dataset to assess transfer: how well the SSL model can adapt its abilities to downstream tasks. Models are assessed by their average transfer learning performance on downstream tasks across domains.

We anticipate a few common questions about DABS:

Why do we need a benchmark for domain-agnostic SSL? Benchmarks catalyze progress by providing a common set of tasks, rules, and evaluation criteria for research towards a particular goal. In this case, DABS provides a standardized way to evaluate the performance of domain-agnostic methods. Fixing the choice and preprocessing of datasets allows for clean comparisons over a range of diverse domains, enabling researchers to pinpoint what specific changes contribute to the success of different methods. Furthermore, the provided infrastructure for data processing, training, transfer, and evaluation significantly reduces the barrier to entry for other researchers interested in these
Figure 2: **Domain-agnostic SSL methods may benefit a wide range of domains.** Self-supervised learning algorithms can provide considerable boosts in performance, but developing them requires considerable effort by machine learning researchers for each domain. Advances in domain-agnostic methods may advance both popular ML domains as well as less studied ones. **Low-ML domains**: algorithms may be used out-of-the-box if no SSL methods exist. **High-ML domains**: algorithms may be used in a hybrid fashion with existing domain knowledge or algorithms. **All domains**: algorithms may provide insights into principles underlying the success of SSL across modalities, accelerating future progress.

**Why might we expect there to be a good domain-agnostic method?** Many kinds of naturally-occurring and artificial data exhibit structure which humans can exploit to learn transferrable skills 
[13, 24, 10, 26]. Human-relevant data (as opposed to white noise) is often generated by some complex generative process. For example, the PAMAP2 wearable sensor dataset [67] is produced by a cascade of latent factors including human interpretation of an activity command, the bodily mechanics of the activity’s execution, and the physical properties of different kinds of sensors that produce measurements. Domain-agnostic pretraining objectives may enable models to capture these latent factors if they are useful for compressing the data (e.g. via density estimation objectives like language modeling [70]) or distinguishing examples from one another (e.g. via contrastive learning objectives [55]). Furthermore, studies on transfer learning of deep networks suggests there exist useful and general “subroutines” learned by SSL models which enable the model to transfer well to new datasets [91, 24]. Empirically, the recent progress of existing domain-agnostic methods [75, 48, 80] is cause for optimism about the future success of this research direction.

**What does domain-agnostic mean?** The goal of DABS is to catalyze the creation of SSL algorithms which are useful out-of-the-box across different domains. We operationalize this goal by evaluating algorithms on a suite of six diverse domains crossing many different fields of machine learning. We also propose several constraints on submissions, described in Section 3, to prevent “overfitting” to these domains. For example, algorithms must use a set of provided dataloaders and keep their architecture and pretraining objective constant across domains (Section 3). However, we also rely on a degree of pragmatism and collaborative ethos from users of DABS to abide by the spirit of the benchmark; for example, a “domain agnostic” algorithm that uses an if-statement to select domain-specific methods for each domain would likely not generalize to new domains. To this end, we also aim to introduce new domains in the future to test the generalizability of proposed algorithms.

To summarize, our **contributions** are:

1. We propose and motivate the task of domain-agnostic self-supervised learning.
2. We present a benchmark for measuring domain-agnostic self-supervised learning, including standardized data loaders and rules for ensuring fair comparisons across submissions.
3. We present two domain-agnostic baseline algorithms and evaluate them on our benchmark, showing relatively modest improvements over baselines that were not pretrained. This suggests ample room for future methods to drive progress.

2 Related Work

Single-domain transfer learning benchmarks Several works have created benchmarks from multiple datasets from a single domain, often with the aim of measuring the “general” understanding capabilities of a single model by measuring its performance across those tasks. Such datasets have been developed in natural language processing \[81, 82\], computer vision \[77, 95, 94\], speech processing \[71, 89\], molecular machine learning \[87\], robotics \[93\], graphs \[69\], and reinforcement learning \[6\], among others.

While these datasets often focus on how a single model can adapt to multiple downstream tasks in a domain, they are typically agnostic to the specifics of the pretraining process—encouraging a “no holds barred” setting where larger models, datasets, and domain-specific assumptions are all utilized to increase downstream accuracy. By contrast, our goal here is to develop general techniques that can be used out-of-the-box for acquiring transferrable capabilities from unlabeled data from any domain. Thus, we hold the pretraining data fixed, allowing researchers to improve only the (domain-agnostic) pretraining algorithm, model architecture, and transfer procedure.

Modality-agnostic architectures In order for an SSL method to be usable out-of-the-box, the model architecture must be applicable in new domains without much customization. Transformers \[79\], originally developed for text, have recently shown promise as a more general architecture for SSL through successful extensions to computer vision \[25\], molecular data \[68\], speech processing \[63\], and multimodal data \[53\] \[24\] \[7\]. These approaches typically use locality assumptions about continuous data (e.g., breaking the input into patches) to map the data into a sequence of embeddings, which are then processed by the transformer. Our baseline algorithms, e-Mix and ShED, leverage similar ideas to train transformer models across all six of our domains, however we expect and encourage future work to explore other flexible architectures, such as the Perceiver \[24\] which relaxes these locality assumptions at the cost of increased computational demands.

Domain-agnostic self-supervised algorithms Several streams of work have recently developed more general SSL methods. Recent work in contrastive learning has sought to reduce the reliance of the objective on domain-specific augmentation functions. The most common approach seeks to find heuristic augmentations which are applicable across a wider range of domains \[48, 80, 92\], while other work seeks to develop generative models which learn distortions during training with a suitable objective \[75\]. Outside of contrastive learning, masked language modeling \[22\] or replaced token detection \[20\] have been applied to other kinds of tokenized or discrete data \[31\] \[60, 13\], but require modification when applied to continuous domains \[25, 53\]. However, none of these algorithms are applicable out-of-the-box on the DABS datasets, so in this work we propose simple domain-agnostic extensions of algorithms in both of these families.

Transfer learning methods Evaluating a self-supervised learning algorithm requires a method to transfer model’s abilities to other tasks of interest \[11, 14, 8\]. These strategies are typically quite domain-agnostic, but involve tradeoffs between various properties, including complexity, downstream performance, and the degree to which they modify the original model. The two most common transfer strategies have historically been training simple linear classifiers on activations extracted from these pretrained models \[24, 58, 63\], or finetuning, where one can often achieve higher performance by specializing the entire model to the downstream task via end-to-end training \[69, 52, 64, 22\]. However, recently, other transfer methods have shown initial success in capturing the benefits of both these extremes, including directly specifying the task in natural language \[12\], as well as approaches that train only a small subset of parameters in the original model \[29, 71\] or that inject trainable features into the input \[47, 54, 49\] or hidden states \[51\] of the model. We permit and encourage users of DABS to investigate different domain-agnostic transfer methods in order to understand their tradeoffs and performance across different domains.
3 Evaluating Domain-Agnostic SSL Algorithms with DABS

How should we evaluate a domain-agnostic SSL method? In DABS, the ultimate goal is to produce a general out-of-the-box solution for SSL across domains—one that generalizes without much modification to arbitrary any desired application. However, one challenge is that SSL methods are comprised of many factors, including the data, pretraining objective, model architecture, and transfer method. Here we describe how the rules of DABS ensure fair comparisons across each of these:

Datasets DABS consists of multiple datasets spread across six domains (detailed in Section 4). To establish fair comparisons across algorithms, we standardize the data loading process, ensuring the same train/test splits, resolutions, tokenizers, and other details. As our primary aim is to measure the performance of methods in a domain-agnostic setting, as opposed to competing with domain-specific methods, we also prohibit the use of data augmentations which vary between domains (e.g. cropping-and-resizing used in natural images). While this may result in lower performance on transfer metrics, past work has shown that domain-specific augmentations can often be integrated into domain-agnostic algorithms to provide gains [75, 53, 80].

Pretraining method The goal of an SSL objective is to enable a model to acquire general capabilities from unlabeled data. To evaluate this, a single pretraining method is used to train a model on each pretraining dataset (Figure 1). Crucially, this method may not be changed by hand between modalities (e.g. adding auxiliary losses for text). However, we do allow adaptive methods that alter the pretraining task in a general way based on the model’s interaction with the unlabeled data, e.g. by learning a generative model to produce input-dependent distortions as in Tamkin et al. [25].

Model architecture While the main model architecture should be kept constant, a key challenge is that different datasets have different input types and dimensions. Some recent works, such as Jaegle et al. [44], attempt to build more data-agnostic model architectures capable of handling different kinds of inputs, however this comes at a significant compute cost. We take a more permissive stance, allowing different types of data to have specialized embedding modules that convert an example from the dataset into a series of vectors. These vectors then serve as the input to a model which is otherwise held constant across the datasets.

We provide a starter set of embedding modules compatible with sequence modeling architectures such as transformers [79]. However, we encourage development of other general input modules as long as their tradeoffs are made clear when comparing against other methods. For continuous rectangular inputs, we use patch embeddings [25] with standardized patch sizes (see Table 2), and for text we use a standard token embedding lookup table, where the text is tokenized using the HuggingFace BertTokenizer. For multimodal data, we apply the appropriate embedding modules to each input, then concatenate the resulting embedding sequences in the same order each time. These embeddings are the only parts of the model which differ across domains, enabling the main architecture to operate identically on each embedding sequence.

Transfer method The ultimate measure of an SSL model is how well it performs when its capabilities are adapted to new tasks. Crucially, transfer methods are distinct from pretraining objectives, and must be compared in their own right as first-class components of an SSL algorithm. Like pretraining techniques, transfer methods exhibit tradeoffs beyond task performance: for example, finetuning a model may produce high accuracy, but requires a separate copy of the model for each use case. Other methods, such as linear evaluation [27], in-context learning [12], and p-, prefix-, and prompt tuning [54, 53, 51] enable the same model to be reused across tasks, but may achieve worse performance in some settings. We allow any transfer method as long as it is held constant across domains and downstream tasks.

Final evaluation metric There are many metrics one might use to compare SSL algorithms, including downstream accuracy, speed, fairness, and cost [28]. In this work, we focus on absolute performance of the model on the given data, for a given number of pretraining steps. However, Note that a transfer method does not presuppose a particular loss function, which may in general vary across tasks. For example, one can finetune a model for both regression and classification tasks.
participants may also be interested in other factors, including compute or data efficiency, or the scaling coefficient of techniques \cite{45,37}. We encourage users of the benchmark to consider any of these, as long as they make clear what previous work is comparable and perform ablations to identify which specific changes impacted the metric being measured.

4 Domains and Datasets

Here, we describe the domains and datasets that comprise DABS. Domains were chosen to span a mix of impactful areas, including domains with both large and small ML research communities. We also aim to include datasets of moderate size, so that researchers with more modest compute budgets can participate. Dataloading and preprocessing within each dataset has been standardized to ensure fair comparisons; more information about data processing for each modality may be found in the Appendix.

Natural images Two-dimensional color images of the natural world is a deeply-studied domain in machine learning. For an unlabeled pretraining dataset, we use CIFAR-10 \cite{47}, a pervasive image classification benchmark in machine learning consisting of 60k images from 10 classes, including cars, frogs and ships. We measure the average transfer accuracy on several image recognition tasks commonly used to assess transfer of pretrained vision models \cite{16,34}: the FGVC-Aircraft dataset \cite{57}, the Caltech-UCSD Birds Dataset \cite{85}, the German Traffic Sign Recognition Benchmark \cite{48}, the Describable Textures Dataset \cite{19} DTD, the VGG Flower Dataset \cite{61}, and the labeled CIFAR-10 dataset itself.

Speech recordings Speech processing is another large community with significant ML presence. We pretrain using the LibriSpeech corpus \cite{62}, a large English-language audiobook corpus commonly used for pretraining. We evaluate transfer to several datasets, including the VoxCeleb \cite{60} and LibriSpeech \cite{62} speaker recognition datasets, and the Fluent Speech Commands \cite{56} action, object, and location], Google Speech Commands \cite{55}, and AudioMNIST \cite{5} utterance classification tasks. To prepare inputs for models, we preprocess examples into log-mel spectrograms—a format which
different significantly from natural image data, and thus may pose challenges for natural image-specific SSL approaches.

**Text** The discrete, tokenized nature of text data makes it very different in form from the two previous continuous domains. For pretraining, we provide WikiText-103 \[59\] a large English-language text dataset collected from Wikipedia. For transfer, we evaluate on the GLUE benchmark \[81\], a suite of natural language tasks including natural language inference, sentiment classification, and paraphrase classification, commonly used to measure transfer of pretrained models to English-language tasks.[4]

**Medical imaging** Medical image understanding encompasses a rich set of domains which often possess ample unlabeled data yet limited labeled data, making them ideal targets for SSL. However, the statistics of medical images can differ significantly from natural images, including lower variation across many inputs and subtler task-relevant features that indicate presence of a pathology \[65\]. However, medical imaging boasts less of an ML presence than natural images, despite the fact that many techniques developed for the former may not apply—e.g., color transformations for black-and-white scans. We focus on chest x-rays as a representative medical imaging domain. We pretrain on the large CheXpert \[42\] dataset of chest x-rays, and assess how well the pretrained model adapts to binary multiclass classification of five pathologies: atelectasis, cardiomegaly, consolidation, edema, and pleural effusion.[5]

**Multi-channel sensor data** Scientific applications are a promising and data-rich area where SSL shows significant promise. However, many scientific domains have a very scarce ML presence compared to domains like natural images. As an example, we consider multi-channel sensor data from wearable devices. We use the PAMAP2 dataset \[67\], consisting of 52-channel sensor recordings (including accelerometer, gyroscope, and magnetometer data) recorded from different body parts as participants perform varied physical activities. We measure transfer to the labeled PAMAP2 task of human-activity recognition: classifying the activity captured in a given recording snippet (e.g. cycling or walking). Examples are processed into 52-channel spectrograms, making this modality different in both shape and content from the single-channel speech spectrograms.

**Captioned images** Multimodal models are an increasingly important area in machine learning. An example domain with two modalities, we consider natural images with paired text captions. We pretrain on image-caption pairs from the COCO dataset \[52\]. We then assess the model’s ability to adapt to the Visual Question Answering task \[2\], reformulated as a binary task to predict whether a question-answer pair correctly describes an image.

### 5 Domain-Agnostic Baseline Algorithms

A domain agnostic self-supervised algorithm is comprised of a domain-agnostic encoder, pretraining objective, and transfer method for learning downstream tasks. However, to the best of our knowledge no previously-proposed method is compatible off-the-shelf with all of the domains in DABS. To establish some baseline approaches, we propose two simple, domain-agnostic algorithms that we evaluate on DABS, which we describe below and hope will serve as useful starting points for future research on domain-agnostic SSL. The core idea behind these algorithms is simple: use a small set of domain-specific embeddings modules to map inputs into an embedding space, and then define the pretraining task directly on those embeddings as opposed to the original inputs.

#### 5.1 Transformer Architecture

Our algorithms use a generalized architecture based on transformers \[72\]. These transformers take as input the sequence of embeddings obtained from the DABS embedding modules, then process them through a series of self-attention and feed-forward layers. We use a 12-layer transformer with hidden size 256, 8 attention heads, and dropout with probability 0.1. To obtain a feature vector for the input, the activations from the final layer are averaged and projected to a 128-dimensional vector.

\[4\] The GLUE benchmark tasks are CoLA \[84\], SST-2 \[72\], MRPC \[23\], QQP \[43\], STS-B \[15\], MNLI \[86, 11\], QNLI \[68, 81\], RTE \[21, 4, 31, 9\], and WNLI \[50\].

\[5\] These are known as the CheXpert “competition tasks” \[42\].
The sequence lengths vary in length depending on the dataset input dimensions and patch sizes, listed in Table 2. The same Transformer architecture is used across all experiments and is optimized with the AdamW optimizer with learning rate 1e-4 and weight decay 1e-4.

5.2 Pretraining Objectives

Given this common architecture, the models are then optimized with respect to a pretraining objective, which enables them to learn useful capabilities and representations from the data. We propose two baseline domain-agnostic SSL objectives, which generalize existing domain-specific methods:

5.2.1 e-Mix: A Contrastive Embedding-Mixup Objective

Contrastive learning and other view-matching objectives have made great strides establishing themselves as competitive or even superior alternatives to supervised pretraining in computer vision.

Sequences are additively mixed with other examples in the dataset. This produces mixed inputs for some random permutation of the input. For a given example, the task is to learn an encoder $f$ such that the vector $f(x_{\pi(i)})$ is close to both $f(x_{i})$ and $f(x_{\pi(i)})$ in proportion to their respective mixing coefficients. Formally, the loss is:

$$
\ell_{e-Mix}(x, \pi, \lambda) = - \sum_{n=1}^{N} v_{i,n} \log \frac{\exp(\text{sim}(f(x_{i}), f(x_{n}))/\tau))}{\sum_{k=1}^{N} \exp(\text{sim}(f(x_{i}), f(x_{k}))/\tau)}
$$

where \text{sim} denotes the cosine similarity, $\tau > 0$ is the temperature, and $v_{i,n}$ is a virtual label given by

$$
v_{i,n} = \begin{cases} 
\lambda, & \text{if } n = i \\
1 - \lambda, & \text{if } n = \pi(i) \\
0, & \text{otherwise}
\end{cases}
$$

The main generalization provided by e-Mix is that the mixup noise is applied to the outputs of the embedding modules (i.e. the patch or token embeddings), as opposed to the inputs directly, which may not in general be continuous. This enables e-Mix to be applied without changes to each of the six domains in the benchmark. To obtain $f(x_{i})$ for a given example $x_{i}$, we process an input with the transformer, then mean pool the outputs along the sequence length dimension, and finally pass the resulting vector through a fully-connected layer with output size 128.

5.2.2 SHED: A Shuffled Embedding Prediction Objective

In contrast to the contrastive objectives that have become common in continuous domains such as images and speech, token-level objectives have been more common in natural language processing. Perhaps the most paradigmatic example is masked language modeling, where random tokens in the input are either redacted or modified and the goal of the main network is to denoise the input.

While this approach has been successfully applied to text, as well as other domains such as images and audio, domain-specific modules are still needed to predict the inputs, which may be downsampling, averaged, or otherwise processed to improve performance.

To avoid this domain-specific complexity, we generalize another family of objectives based on the ELECTRA method for pretraining on text. Rather than reconstruct noised tokens as BERT does, ELECTRA involves replacing a subset of tokens in the input with substitutes, then training a detector network to predict which tokens were replaced. Substitute tokens can be chosen randomly, or generated by a BERT network. Similar replacement-detection methods have also recently been applied successfully to tabular data, suggesting this objective is not text-specific.
<table>
<thead>
<tr>
<th>Pretraining</th>
<th>Natural Images</th>
<th>Text</th>
<th>Speech</th>
<th>Sensors</th>
<th>Med. Imaging</th>
<th>Captioned Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>10.1</td>
<td>42.3</td>
<td>24.9</td>
<td>93.6</td>
<td>68.1</td>
<td>57.5</td>
</tr>
<tr>
<td>e-Mix</td>
<td>27.9</td>
<td>44.1</td>
<td>41.8</td>
<td>91.4</td>
<td>72.4</td>
<td>48.9</td>
</tr>
<tr>
<td>ShED</td>
<td>20.9</td>
<td>48.4</td>
<td>36.5</td>
<td>91.2</td>
<td>74.5</td>
<td>54.3</td>
</tr>
</tbody>
</table>

Table 1: Downstream linear classifier performance of baseline domain-agnostic methods across domains. Reported numbers are average evaluation metrics across transfer tasks within a domain. Metrics are percent accuracy, with the exception of Medical Imaging (average percent AUC across the five pathologies), and two Text tasks: CoLA (Pearson correlation) and STS-B (Spearman correlation). “None” refers to a randomly-reinitialized model that has not been trained.

We generalize ELECTRA by applying replacements at the embedding level, instead of the level of input tokens. This enables us to apply the method equally across all modalities, without domain-specific adjustments. To perform the replacements, we select 15% of the embedding positions per input, then shuffle those embeddings among each other according to a random permutation. See the Appendix for more details. The task of the network is then to predict which of the embeddings were shuffled; this is instantiated as a binary prediction task performed by passing each output embedding through a fully-connected layer. We call this method ShED: Shuffled Embedding Detection.

5.3 Linear Classification

We transfer our trained models to downstream tasks with linear classifiers, a simple approach which enables the same base model to be adapted to many downstream tasks without storing a separate copy of the model for each task. We use the Adam optimizer \[^46\] with learning rate of 1e-4, \(\beta_1 = 0.9, \beta_2 = 0.999\) for 100 epochs. We also compare against a randomly-initialized model which has not undergone training, to quantify the gains attributable to pretraining.

5.4 Results

We report average metrics by domain in Table 1 and full results for each transfer task in Table 3. Our pretrained models broadly show gains over models that were not pretrained, although the gains are uneven and often quite modest compared to state-of-the-art domain-specific approaches. While the gains from pretraining are clear across transfer tasks in natural images, speech, text, and medical imaging, pretraining appears to hurt in sensor and captioned image domains, leaving a clear need for future work. Interestingly, the relative gains for these algorithms also seems to reflect their communities of origin: e-Mix performs best on natural images, while ShED performs better on text-based tasks. Investigating the principles underlying these differences is an interesting avenue for future work, as is discovering methods that work better across all domains.

6 Limitations and Conclusion

We have proposed DABS: a Domain-Agnostic Benchmark for Self-Supervised Learning. Algorithms that perform well on DABS may have significant practical impact, unlocking the benefits of pretraining for a wide array of domains without a significant ML presence. We also hope DABS encourages researchers to investigate the general principles underlying self-supervised pretraining objectives, model architectures, and transfer methods.

DABS also has limitations. For example, a tradeoff exists between keeping a benchmark reasonably compact so it can be run easily and representing the full range of domains one might care about. Our choice of six diverse domains represents a middle ground, but DABS is also a “living benchmark,” and we plan in the future to introduce domains spanning an even broader range of fields, data types, and applications to drive further progress towards domain-general SSL methods.

In addition, DABS does not capture how well domain-agnostic methods can be combined with domain-specific methods in a hybrid manner, which may be of greater relevance to domains like natural images where many domain-specific augmentations have already been developed. This is an important yet challenging-to-frame problem, and we encourage future work in this direction.
References


Checklist

1. For all authors...

(a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes] Yes, we motivate the need for DABS in Section 1, we describe the benchmark itself in Sections 3 and 4, and we present baseline algorithms in Section 5.

(b) Did you describe the limitations of your work? [Yes] Yes, in Section 6.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] Yes, in Section F.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [N/A] DABS did not involve the collection of new datasets.

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments (e.g. for benchmarks)...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Yes, our repository with detailed instructions in the README will be made public on GitHub and has been uploaded for the reviewers during the review phase.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Yes, in Sections 4, 5, and G and in Table 2.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] The total number of runs required was 85, including 12 pretraining runs, so we were unable to run many trials for each run.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Yes, in Section B.

4. If you are using existing assets (e.g., code, data, models) or curating/releases new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Yes, we cite the creators of each dataset in Section 4.

(b) Did you mention the license of the assets? [Yes] Yes, in Section C.

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A] DABS did not involve the collection of new datasets.

(d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes] Yes, in Section D.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Yes, in Section E.
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]