Unsupervised Embedding Quality Evaluation

Anton Tsitsulin [1] **Marina Munkhoeva** [2] **Bryan Perozzi** [1]

Abstract

Unsupervised learning has recently significantly gained in popularity, especially with deep learning-based approaches. Despite numerous successes and approaching supervised-level performance on a variety of academic benchmarks, it is still hard to train and evaluate SSL models in practice due to the unsupervised nature of the problem. Even with networks trained in a supervised fashion, it is often unclear whether they will perform well when transferred to another domain.

Past works are generally limited to assessing the amount of information contained in embeddings, which is most relevant for self-supervised learning of deep neural networks. This works chooses to follow a different approach: can we quantify how easy it is to linearly separate the data in a stable way? We survey the literature and uncover three methods that could be potentially used for evaluating quality of representations. We also introduce one novel method based on recent advances in understanding the high-dimensional geometric structure of self-supervised learning.

We conduct extensive experiments and study the properties of these metrics and ones introduced in the previous work. Our results suggest that while there is no free lunch, there are metrics that can robustly estimate embedding quality in an unsupervised way.

1. Introduction

With proliferation of unsupervised and self-supervised deep learning methods in the recent years, there is an increasing need to quantify the quality of representations produced by such methods. Across different domains, this is commonly done with training linear classifiers (*probes*) against known labels (Perozzi et al., 2014; Chen et al., 2020). However, in unsupervised settings *there are no labels* to begin with. How can we do model selection, optimize methods' hyperparameters, or even verify the method worked at all?

In search of such metrics, we turn our attention to different sub-fields of numerical linear algebra, machine learning and optimization, and high-dimensional probability. We identify three promising candidate metrics and introduce one based on the expected distribution of embedding distances. We then proceed to test them on two conceptually novel domains: *supervised* model selection and shallow single-layer graph embedding learning.

Our experimental results indicate there is no "free lunch"—a metric that is universally dominating—thus calling for a comprehensive suite of evaluation metrics. Despite that, metrics introduced in this work exhibit, like stable rank and coherence, display stronger correlation to downstream task performance of the supervised models, are more computationally stable, and suit shallow embedding models much better than state-of-the-art ones.

We summarize our key contributions as follows:

- We identify three different perspectives on evaluation of embedding quality in unsupervised manner and introduce four metrics based on these perspectives.
- We experimentally study two novel settings for embedding quality evaluation, showing that standard metrics often fail when shallow models are being studied.
- We conduct a study on computational stability of all metrics and identify the minimum viable sample sizes.
- We demonstrate that the proposed metrics are at least as effective as state-of-the-art ones in terms of downstream quality prediction while having more intuitive behavior for shallow embedding models.

2. Related Work

The literature on evaluating representations in unsupervised way is still sparse. Arguably, *dimensional collapse* (Hua et al., 2021) has sparked initial interest in the area. In dimensional collapse, some dimensions become non-meaningful (collapse) during training. Because of that problem, three concurrent metrics, which we introduce below, all study the problem of measuring such collapse from different angles.

[1]Google Research, New York, USA [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany. Correspondence to: Anton Tsitsulin <tsitsulin@google.com>.

Proceedings of the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

α-ReQ (Agrawal et al., 2022) fits a power-law to the singular values of representations, meaning $\lambda_i \propto i^{-\alpha}$. Logarithmic decay of the spectrum with slope $\alpha = 1$ was recently proven to provide the best generalization in infinite-dimensional analysis of linear regression (Bartlett et al., 2020). In practice, a simple linear regression estimator on a log-log scale is used to estimate the value of α. This approach for estimating the power-law exponent is considered inaccurate (Clauset et al., 2009).

RankMe (Garrido et al., 2022; Roy & Vetterli, 2007) is a method based on estimating the effective rank of a matrix. In a strict numerical linear algebraic sense, most embedding matrices are full-rank. "Softer" definitions allow to capture not only fully collapsed dimensions but also general underutilization of the parameter space.

Definition 2.1. Given a matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ with SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, its effective rank is the entropy of its normalized singular values, defined as

$$\text{RankMe}(\mathbf{M}) = -\sum_i p_i \log p_i, \quad p_i = \frac{\sigma_i}{\|\mathbf{\Sigma}\|_1}.$$

NESum (He & Ozay, 2022) analyzes eigenspectrum of the covariance matrix of representations. It is introduced as a heuristic metric complementing the analysis of features learned by the barlow twins loss (Zbontar et al., 2021).

Definition 2.2. Given a matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ with covariance that can be decomposed as $\mathbf{C} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$:

$$\text{NESum}(\mathbf{M}) = \sum_i \frac{\lambda_i}{\lambda_0},$$

with convention of $\frac{0}{0} = 0$.

3. Three Perspectives on Embedding Quality

We now study three different perspectives on estimating embedding "quality". All measures we have discussed so far aim to answer an information-theoretic question on representations: *Do embedding carry as much information as their size allows?* However, there are different questions worth answering. This paper introduces four novel metrics for embedding quality evaluation based on different perspectives on the embedding quality.

The following section pursues the linear classifier perspective on representation quality (Mohri & Talwalkar, 2011). It asks: *How hard it is to find a suitable transformation from the representations to the targets of the downstream task?* We show that this is an inherent property of the representations themselves (and the target matrix too, if it's not a classification task).

3.1. Linear Classifier Perspective

Let our downstream task be a classification with a target matrix $\mathbf{Y} \in \{0,1\}^{n \times c}$ and a linear probe $h = \mathbf{X}\mathbf{W} + \mathbf{b}$ with weight matrix $\mathbf{W}$ and bias vector $\mathbf{b}$. In what follows, we argue that it is easier to find h that yields high accuracy when applied to the input matrix $\mathbf{X}$ with higher coherence.

Without loss of generality, we can drop the bias term. For the ease of exposition, we will adopt the Mean-Squared Error loss ($\mathcal{L} = ||\mathbf{Y} - \mathbf{X}\mathbf{W}||_F^2$) for a downstream task. The optimal weight matrix will then depend on the target and representation matrices, i.e. from the derivative condition $\mathbf{X}^\top\mathbf{Y} = \mathbf{X}^\top\mathbf{X}\mathbf{W}$. Given some $\mathbf{A} \in \ker(\mathbf{X})$, i.e. a matrix comprised of vectors from the null space of $\mathbf{X}$, we rewrite the condition as $\mathbf{X}^\top\mathbf{Y} = \mathbf{X}^\top(\mathbf{A} + \mathbf{X}^\dagger\mathbf{Y})$ and get $\mathbf{W}^* = \mathbf{X}^\dagger\mathbf{Y} + \mathbf{A}$ for any $\mathbf{A} \in \ker(\mathbf{X})$.

Assuming we can always find an optimal weight matrix, to minimize the loss $\mathcal{L}$, the representations $\mathbf{X}$ should be aligned with the target matrix $\mathbf{Y}$, i.e. the left singular vectors $\mathbf{U}$ of $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ should span $\mathbf{U}_\mathbf{Y}$ of $\mathbf{Y} = \mathbf{U}_\mathbf{Y}\mathbf{\Sigma}_\mathbf{Y}\mathbf{V}_\mathbf{Y}$, where $\mathbf{V}_\mathbf{Y} = \mathbf{I}_c$ when $\mathbf{Y}$ is a classification target matrix.

Plugging in the optimal $\mathbf{W}^*$ into the loss,

$$\begin{aligned}
||\mathbf{Y} - \mathbf{X}(\mathbf{X}^\dagger\mathbf{Y} + \mathbf{A})||_F^2 &= \|\mathbf{Y} - \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^\dagger\mathbf{U}^\top\mathbf{Y}\|_F^2 \\
&= \|(\mathbf{I} - \mathbf{U}\mathbf{I}_d\mathbf{U}^\top)\mathbf{Y}\|_F^2 \\
&= \|(\mathbf{I} - \mathbf{I}_d)\mathbf{U}^\top\mathbf{Y}\|_F^2 \\
&= \|\mathbf{Y}\|_F^2 - \|\mathbf{U}_d^\top\mathbf{U}_\mathbf{Y}\mathbf{\Sigma}_\mathbf{Y}\|_F^2,
\end{aligned}$$

where $\mathbf{I}_d \in \mathbb{R}^{n \times n}$ with d ones on the diagonal, and the minimum is reached whenever columns in $\mathbf{U}$ are aligned with columns in $\mathbf{U}_\mathbf{Y}$.

Intuitively, if the representation dimensionality is larger than number of classes in the downstream task, i.e. $d > c$, and $\mathbf{X}$ has full rank (a consequence of most methods being spectral embedding), then the representation basis covers the target basis with high probability. However, to quantify the extent of this coverage, we will need to introduce a notion of incoherence.

Definition 3.1 (μ_0-incoherence)**.** Given matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ with rank-r and SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, $\mathbf{M}$ is said to satisfy the *standard incoherence* condition with parameter μ_0 if

$$\max_{1 \le i \le n_1} ||\mathbf{U}^\top e_i||_2 \le \sqrt{\frac{\mu_0 r}{n_1}}, \quad \max_{1 \le i \le n_2} ||\mathbf{V}^\top e_j||_2 \le \sqrt{\frac{\mu_0 r}{n_2}},$$

where e_i is the i-th standard basis vector of a respective dimension. Note that $1 \le \mu_0 \le {}^{\max(n_1,n_2)}/_{r}$.

Informally, standard incoherence characterizes the extent of alignment of the singular vectors to the standard basis.

Incoherence is typically used in low-rank matrix completion problems to estimate a complexity of matrix recovery (Mohri & Talwalkar, 2011). In our setting, *lower* incoherence will be indicative of high alignment with target matrix and, thus, *better* performance.

Ideally, if we had access to the targets, we could use joint incoherence $\mu_1(\mathbf{Z}, \mathbf{Y})$ to measure the alignment directly. More practical is the case when true labels are not available. There, we will need to rely on the standard coherence $\mu_0(\mathbf{Z})$ which measures alignment to the standard basis. Our experiments show that there is indeed a correlation between standard incoherence of the representations and performance on the downstream tasks (almost perfect in some cases).

3.2. Numerical Linear Algebra Perspective

Numerical linear algebra provides us with more tools for analysing behaviors of linear classifiers. One of the classic ones is the condition number, or, in the case of non-square matrices, its generalized version (Ben-Israel, 1966). For example, κ_2 is used to detect multicollinearity in linear and logistic regression (Belsley et al., 2005).

Definition 3.2. Pseudo-condition number of a matrix $\mathbf{M}$ with SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is defined as

$$\kappa_p(\mathbf{M}) = \|\mathbf{M}\|_p \|\mathbf{M}^\dagger\|_p \overset{p=2}{=} \frac{\sigma_1}{\sigma_n}.$$

We are particularly interested in κ_2, since it is easily computable with SVD, as the pseudo-inverse of $\mathbf{M}$ is $(\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M} = \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{V}^\top$, meaning $\|\mathbf{M}^\dagger\|_2 = {}^1/_{\sigma_n}$.

In the analysis of linear regression, κ_2 can be used to bound the sensitivity of the system to the change in the input. Consider a linear system $(\mathbf{A} + \Delta\mathbf{A})\hat{\mathbf{x}} = \mathbf{b}$ and its perturbed version $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b} + \Delta\mathbf{b}$. Then,

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \le \frac{\kappa(\mathbf{A})}{1 - \kappa(\mathbf{A})\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}} \left(\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|} + \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} \right).$$

We use κ_2 to measure stability of learned representations.

3.2.1. Stable Rank

Stable rank (also called *effective* rank or intrinsic dimension of a matrix) is another fundamental quality in numerical analysis of random matrices.

Definition 3.3. Numerical rank of a matrix $\mathbf{M}$ is defined as

$$r(\mathbf{M}) = \frac{\|\mathbf{M}\|_F}{\|\mathbf{M}\|_2^2}$$

Note that $r(\mathbf{M}) \le \operatorname{rank}(\mathbf{M})$, and that bound is sharp. Stable rank is a useful tool that guides fundamental numerical problems, including matrix sampling and covariance estimation.

Let us restate Theorem 1.1 from Rudelson & Vershynin (2007):

Theorem 3.4. *Let $\mathbf{A}$ be an $n \times d$ matrix with stable rank r. Let $\varepsilon, \delta \in (0, 1)$, and let $m \le n$ be an integer such that*

$$m \ge C\left(\frac{r}{\varepsilon^4\delta}\right)\log\left(\frac{r}{\varepsilon^4\delta}\right).$$

Consider a $m \times d$ matrix $\tilde{\mathbf{A}}$, which consists of m normalized rows of $\mathbf{A}$ picked independently with replacement, with probabilities proportional to the squares of their Euclidean lengths. Then with probability at least $1 - 2\exp(-{}^c/_\delta)$ the following holds. For a positive integer k, let $\mathbf{P}_k$ be the orthogonal projection onto the top k left singular vectors of $\tilde{\mathbf{A}}$. Then,

$$\|\mathbf{A} - \mathbf{A}\mathbf{P}_k\| = \sigma_{k+1}(\mathbf{A}) + \varepsilon\|\mathbf{A}\|_2.$$

This suggests that the numerical rank determines how hard it is to estimate the matrix by subsampling its rows. Intuitively, a well-distributed representations should be hard to estimate; we will observe that this is indeed the case in practice.

3.3. High-dimensional Probability Perspective

In self-supervised learning, Assran et al. (2023) shows that several contrastive learning methods try to distribute representations equally in the space. High-dimensional probability can provide us with an estimate of pairwise distances when embeddings are distributed uniformly on a d-dimensional unit sphere $\mathbb{S}^d$.

Given L_2 normalized embeddings $\mathbf{W} \in \mathbb{R}^{n \times d}$, a measure of clustering can be defined using the norm of the pairwise dot product matrix $Q = \|\mathbf{W}\mathbf{W}^\top\|_F$. Since the expected dot product of high-dimensional isotropic random vectors $\langle \mathbf{x}, \mathbf{y} \rangle \asymp \frac{1}{n}$ (Vershynin, 2018, Remark 3.2.5), we can estimate $\mathbb{E}[Q] = n + \frac{n(n-1)}{d}$. The maximum metric value $Q = n^2$ can only be achieved in the collapsed case. Combining all normalizations to get a metric upper-bounded that is upper-bounded by 1, we get:

Definition 3.5.

$$\begin{aligned}\operatorname{SelfCluster}(\mathbf{W}) &= \frac{\|\mathbf{W}\mathbf{W}^\top\|_F - n - \frac{n*(n-1)}{d}}{n^2 - n - \frac{n*(n-1)}{d}} \\ &= \frac{d\|\mathbf{W}\mathbf{W}^\top\|_F - n(d+n-1)}{(d-1)(n-1)n}.\end{aligned}$$

SelfCluster allows us to estimate how much the embeddings are clustered in the embedding space compared to random distribution on a sphere. The downside of this metric is the requirement of pairwise computations, which is expensive for large number of points. We now proceed to study the proposed metrics on real-world data.

4. Experiments

In contrast to previous work (Agrawal et al., 2022; Garrido et al., 2022), we shift our attention from self-supervised learning to novel, more generally applicable settings. We experimentally study proposed metrics on two novel use-cases: (i) supervised representation learning with deep neural networks and (ii) unsupervised graph embeddings. Supervised representation learning allows us to gain insights into performance of semi-supervised learning systems. Graph embedding, on the other hand, has very different architecture—shallow single-layer network—and optimization.

Section 4.1.2 further provides a novel study on computational stability of different embedding quality evaluation metrics. Stability is important for many practical application, since the most computationally stable metrics can be even computed during training for monitoring purposes.

4.1. Supervised Network Performance Prediction

We used Wightman (2019) repository of supervised PyTorch models, accessed May 2023. (Deng et al., 2009) We ran inference of all available models, as permitted by GPU memory, on the validation set, and a subset of models[1]—on the full training set. Inference was performed on a single 16-core machine with NVIDIA RTX 4090 and 64Gb RAM.

4.1.1. Downstream Quality Correlation

Figures 1 and 2 present rank correlation of the different embedding quality metrics to downstream prediction quality on ImageNet, measured for training and validation set embeddings respectively. We do not report SelfCluster metric results on the training set because of its quadratic time complexity. Since RankMe is dependent on the dimensionality of the data, we normalize its values and call the metric RankMe*. This new metric has the range between 0 and 1, and represents relative utilization of the embedding space.

On the training set evaluation, α-ReQ, NESum, pseudo-condition number, and coherence all show significant correlation to the test set performance. Out of these metrics, α-ReQ is the only metric with significant outliers, possibly due to the power law estimation issues (Clauset et al., 2009). High stable rank, NESum, and coherence seem to indicate good test test performance of the model. Note that the models we selected for training set evaluation are pareto-optimal in terms of either parameter size or inference speed. This allowed us to significantly restrict the model set size without affecting representativeness of selected models.

On the validation set performance with expanded model set, the correlation between many metrics and test set performance drops to near-zero. This can be attributed to both

[1] Full list available in the Appendix.

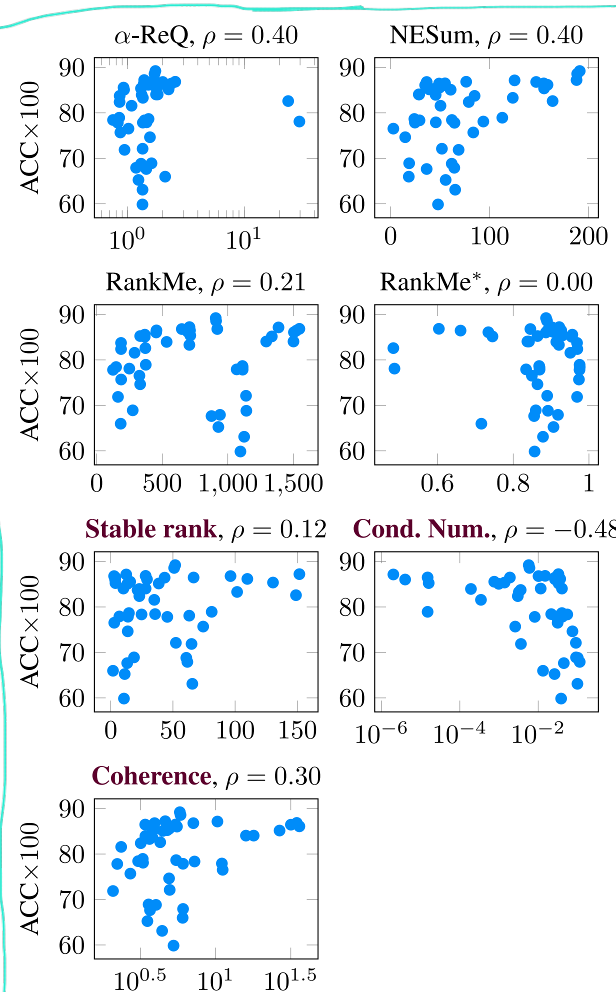

Figure 1. Representation quality metrics on the ImageNet training set for over 30 pre-trained models. Spearman rank correlation ρ to the test set accuracy displayed per metric in the title. Methods introduced in this work are highlighted in **colored bold**.

expanded model set, which has many under-performing models as well as the general instability of the computation on the smaller example set. We further examine the computational stability considerations in the next section. Only NESum, stable rank and self clustering achieve significant correlation to the test set performance. Across both training and validation sets, NESum demonstrates strong downstream performance correlation while both variants of RankMe are not able to successfully predict supervised task performance.

4.1.2. Metric Stability

It is important to have stable metrics for embedding quality evaluation, especially in low-data regimes. Moreover, if a metric is stable up to very small batch sizes, it can be evaluated during training, greatly enhancing its usability.

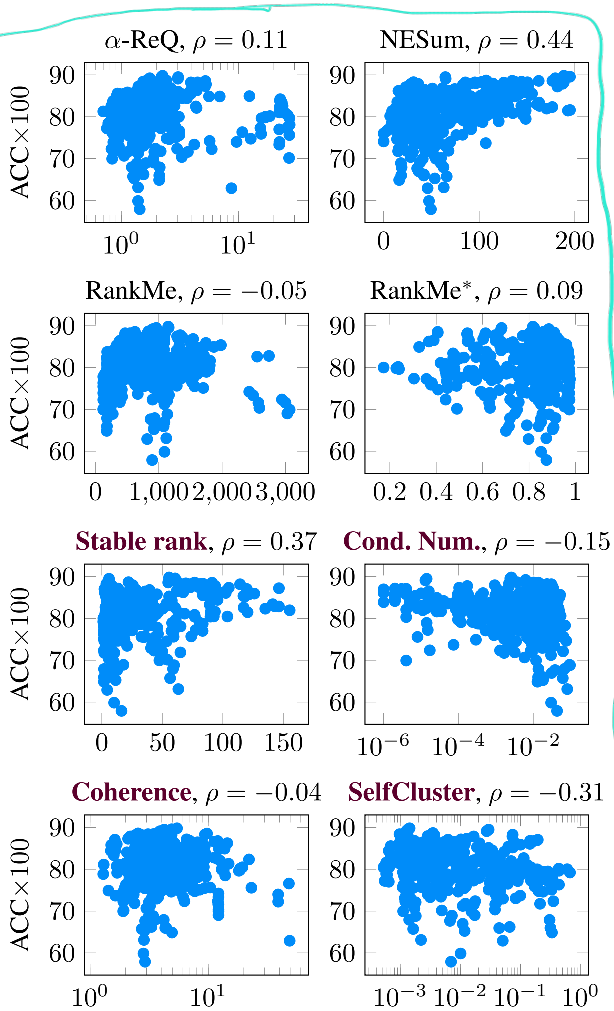

Figure 2. Representation quality metrics on the ImageNet validation set of over 1000 pre-trained models. Spearman rank correlation ρ to the test set accuracy displayed per metric in the title. Methods introduced in this work are highlighted in **colored bold**.

To do that, we sample embeddings for ImageNet training set with batch sizes from 128 to 65536, log-space (2^7–2^{16}) and compare the sampled metric value to the value computed on the whole dataset. The results are presented in Table 1. Numerical rank-based methods are among the most stable, followed by NESum. One advantage of RankMe over its numerical rank estimation counterpart is that it offers a strong lower-bound in terms of the sample size. Coherence appears to be strongly data-dependent and least stable.

4.2. Graph Embedding Quality Prediction

Graph embedding is a common way to solve many tasks arising in the graph mining domain from node classification, link prediction, and community detection. In the graph embedding process, each node in a graph is mapped to a vector in $\mathbb{R}^d$, and distances in the embedding space should resemble some similarity metric defined between the nodes in the original graph (Tsitsulin et al., 2018). For an in-depth review of modern graph embedding approaches, readers are referred to Chami et al. (2022) survey.

Table 1. Batch sizes needed to achieve constant multiplicative approximation factors compared to evaluation on the full ImageNet training set on XX networks. Additionally, we check that each metric lower-bounds the true value. The result can be either ✓yes, ✗no, or ❀0.95-approximately.

		Approximation factor			
metric	Bounded	0.5	0.7	0.9	0.95
α-ReQ	✗	512	4096	32768	—
NESum	❀	1024	2048	8192	32768
RankMe	✓	2048	2048	8192	16384
Stable rank	❀	512	2048	8192	16384
Cond. number	✗	4096	4096	32768	65536
Coherence	✓	—	—	—	—

Table 2. Dataset statistics. We report total number of nodes $|V|$, average node degree $\bar{d}$, number of labels $|Y|$.

dataset	$\lvert V\rvert$	$\bar{d}$	$\lvert Y\rvert$
Cora	19793	3.20	7
Citeseer	3327	1.37	6
PubMed	19717	2.25	3
Amazon PC	13752	17.88	10
Amazon Photo	7650	15.57	8
MSA-Physics	34493	7.19	5
OGB-arXiv	169343	6.84	40
CIFAR-10	50000	99	10
MNIST	60000	99	10

For our experiments, we study representations of the DeepWalk (Perozzi et al., 2014) model as it is a de-facto standard in the field of unsupervised embedding of graphs with no features. We use 10 different graph datasets that include both natural and constructed graphs. We report the dataset statistics in Table 2 and provide a brief description below:

- Cora, Citeseer, and Pubmed (Sen et al., 2008) are citation networks; nodes represent papers connected by citation edges; features are bag-of-word abstracts, and labels represent paper topics. We use a re-processed version of Cora from (Shchur et al., 2018) due to errors in the processing of the original dataset.
- Amazon {PC, Photo} (Shchur et al., 2018) are two subsets of the Amazon co-purchase graph for the computers and photo sections of the website, where nodes represent goods with edges between ones frequently purchased together; node features are bag-of-word reviews, and class labels are product category.
- OGB-ArXiv (Hu et al., 2020) is a paper co-citation dataset based on arXiv papers indexed by the Microsoft Academic graph. Nodes are papers; edges are citations, and class labels indicate the main category of the paper.

Table 3. Average Spearman rank correlation on two dataset corruption types: naïve (N) and component-preserving (C). We highlight datasets where there is a consistent correlation pattern, meaning the same sign and approximately the same magnitude of correlation. Methods proposed in this work exhibit stronger and more consistent correlation patterns across all datasets.

	Cora		Citeseer		Pubmed		Amazon PC		Amazon Photo	
metric	N	C	N	C	N	C	N	C	N	C
α-ReQ	-1.00	-1.00	-1.00	-1.00	-1.00	0.43	0.01	0.98	0.01	0.97
NESum	1.00	0.03	1.00	0.10	0.94	-0.66	0.09	-1.00	-0.15	-1.00
RankMe	1.00	1.00	1.00	1.00	1.00	-0.37	-0.05	-0.99	-0.43	-0.99
Stable rank	1.00	0.66	1.00	0.30	1.00	0.66	0.31	-1.00	0.09	-1.00
Cond. number	1.00	0.83	1.00	1.00	1.00	0.26	0.20	-0.99	0.10	-1.00
SelfCluster	-1.00	-1.00	-1.00	-0.60	1.00	1.00	1.00	0.99	1.00	1.00
Coherence	1.00	1.00	0.90	1.00	0.94	1.00	0.99	0.98	0.99	0.98

	MSA-Physics		OGB-arXiv		MNIST		CIFAR-10	
metric	N	C	N	C	N	C	N	C
α-ReQ	-0.70	0.94	-0.81	1.00	-1.00	0.98	0.96	0.99
NESum	0.51	-0.98	0.84	-1.00	0.99	-0.92	-0.84	-0.99
RankMe	0.59	-0.92	0.85	-1.00	1.00	-0.96	-0.94	-1.00
Stable rank	0.52	-0.97	0.99	-0.99	1.00	-0.78	-0.85	-0.99
Cond. number	0.52	-0.97	0.92	-1.00	1.00	-0.96	-0.95	-0.99
SelfCluster	0.96	0.98	1.00	1.00	1.00	1.00	1.00	0.99
Coherence	0.97	0.99	0.90	1.00	0.89	1.00	0.98	0.99

- CIFAR and MNIST (Krizhevsky et al., 2009; LeCun et al., 1998) are ε-nearest neighbor graphs with ε such that the average node degree is 100.

Instead of changing the parameters of the model, we controllably change the quality of data itself. We sparsify each graph in two different ways:

- **Naïve sparsification**: we randomly pick $n\bar{d}$ edges from the original edge set. This method may produce disconnected components, which are known to be difficult to embed correctly.
- **Component-preserving sparsification**: we first ensure the resulting graph is connected by sampling a random spanning tree. Then, we sample $n(\bar{d}-1)$ edges randomly and output the combined graph.

It is easy to see both versions create a controllably worse version of the data. As such, one could expect that representation quality degrades with the sparsity of the input graph, perhaps faster for the naïve algorithm, since it does not preserve the component information. As we will observe later, surprisingly, this is very much not the case for many embedding quality metrics we study.

We sparsify to a fixed number of edges corresponding to a target average node degree from the range $[1.1, 10]$. Some graphs in our studies have an average node degree < 10 naturally (cf. Table 2), in this case, we stop at that number. We embed each graph 10 times, run a downstream node classification 100 times, and average the result. We report Spearman rank correlation coefficient ρ (Spearman, 1904) between the classification accuracy and each quality metric.

Table 4. Average Spearman rank correlation on two dataset corruption types: naïve and component-preserving. We highlight rows where there is a consistent correlation pattern. Two methods introduced in this work strongly and consistently correlate with the downstream classification performance.

metric	Naïve	Connected
α-ReQ	-0.50	0.48
NESum	0.49	-0.71
RankMe	0.45	-0.47
Stable rank	0.56	-0.46
Cond. number	0.53	-0.43
SelfCluster	0.55	0.60
Coherence	0.95	0.99

First, we report aggregated results across all datasets in Table 4. Surprisingly, most metrics completely revert the correlation sign between two sparsification strategies. Only SelfCluster and Coherence are aligned with the downstream evaluation, and between them, Coherence displays a near-perfect correlation with the downstream task performance.

Table 3 provides a more nuanced per-dataset view. We can observe that while some metrics have strong and consistent correlation patterns on some datasets, the trend can be completely reversed on others. This calls for more comprehensive evaluations on multiple datasets and machine learning tasks for embedding quality evaluation metrics. Overall, only coherence provides strong signal in a single direction across all the datasets and perturbation methods.

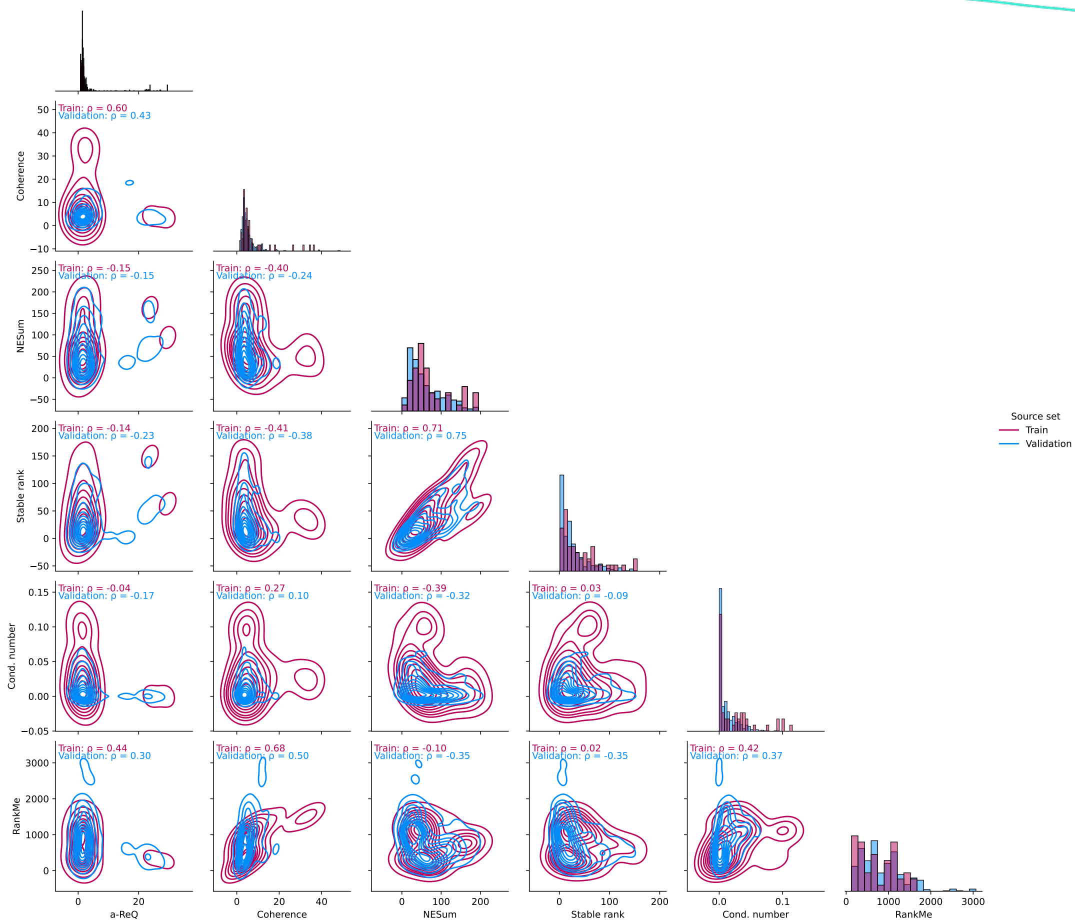

Figure 3. Pairwise density plots of ImageNet representations, as measured on training and validation sets. NEsum is well-correlated to Stable rank. Coherence is moderately correlated to α-ReQ and RankMe.

4.3. Metric Similarity

Since there are no clear winners in the experiments, it is important to use multiple metrics in real-world applications. Figure 3 presents pairwise correlations and kernel densities of different metrics on the training and validation sets of ImageNet. Overall, there are two clusters of the metrics: NESum and Stable rank as one and Coherence, α-ReQ, RankMe and condition number in another.

5. Conclusions

Is it possible to estimate embedding quality based on its statistical properties? This paper demonstrates it is possible in two scenarios outside of the known one of self-supervised learning. We introduced four new metrics based on ideas from numerical linear algebra, analysis of linear regression and high-dimensional probability.

We conducted a large-scale study on two novel domains for unsupervised embedding quality evaluation: prediction of supervised test set performance and predicting performance of much simpler single-layer graph embedding methods. In case of supervised models, there seem to be no one-size-fits-all dominant solution, however, we identify numerically stable metrics that have strong correlation with downstream task performance. In the shallow model case, metrics introduced in this work show favorable downstream performance correlation consistently across 9 different datasets.

References

Agrawal, K. K., Mondal, A. K., Ghosh, A., and Richards, B. α-ReQ: Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. *NeurIPS*, 2022. Cited on pages 2 and 4.

Assran, M., Balestriero, R., Duval, Q., Bordes, F., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., and Ballas, N. The hidden uniform cluster prior in self-supervised learning. In *ICLR*, 2023. Cited on page 3.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. Benign overfitting in linear regression. *PNAS*, 2020. Cited on page 2.

Belsley, D. A., Kuh, E., and Welsch, R. E. *Regression diagnostics: Identifying influential data and sources of collinearity*. John Wiley & Sons, 2005. Cited on page 3.

Ben-Israel, A. On error bounds for generalized inverses. *SIAM Journal on Numerical Analysis*, 1966. Cited on page 3.

Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C., and Murphy, K. Machine learning on graphs: A model and comprehensive taxonomy. *JMLR*, 2022. Cited on page 5.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. Cited on page 1.

Clauset, A., Shalizi, C. R., and Newman, M. E. Power-law distributions in empirical data. *SIAM review*, 2009. Cited on pages 2 and 4.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. Cited on page 4.

Garrido, Q., Balestriero, R., Najman, L., and Lecun, Y. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. *arXiv preprint arXiv:2210.02885*, 2022. Cited on pages 2 and 4.

He, B. and Ozay, M. Exploring the gap between collapsed & whitened features in self-supervised learning. In *ICML*, 2022. Cited on page 2.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020. Cited on page 5.

Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., and Zhao, H. On feature decorrelation in self-supervised learning. In *CVPR*, 2021. Cited on page 1.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009. Cited on page 6.

LeCun, Y., Cortes, C., and Burges, C. J. C. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. Cited on page 6.

Mohri, M. and Talwalkar, A. Can matrix coherence be efficiently and accurately estimated? In *AISTATS*, 2011. Cited on pages 2 and 3.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *KDD*, 2014. Cited on pages 1 and 5.

Roy, O. and Vetterli, M. The effective rank: A measure of effective dimensionality. In *European signal processing conference*. IEEE, 2007. Cited on page 2.

Rudelson, M. and Vershynin, R. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 2007. Cited on page 3.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 2008. Cited on page 5.

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018. Cited on page 5.

Spearman, C. The proof and measurement of association between two things. 1904. Cited on page 6.

Tsitsulin, A., Mottin, D., Karras, P., and Müller, E. Verse: Versatile graph embeddings from similarity measures. In *WWW*, 2018. Cited on page 5.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018. Cited on page 3.

Wightman, R. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. Cited on page 4.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021. Cited on page 2.

A. Appendix.

Here we present the list of models we used for experimenting on the training and validation sets of ImageNet.

Training set models

```
beitv2_base_patch16_224.in1k_ft_in22k_in1k
coat_tiny
convnext_base.fb_in22k_ft_in1k_384
convnext_femto_ols.d1_in1k
dla46x_c
edgenext_base
edgenext_small
edgenext_x_small
edgenext_xx_small
eva_giant_patch14_560.m30m_ft_in22k_in1k
eva_large_patch14_196.in22k_ft_in22k_in1k
eva_large_patch14_336.in22k_ft_in22k_in1k
lcnet_050.ra2_in1k
lcnet_075.ra2_in1k
lcnet_100.ra2_in1k
levit_128s
maxvit_base_tf_512.in21k_ft_in1k
maxvit_large_tf_512.in21k_ft_in1k
mobilenetv3_large_100.miil_in21k_ft_in1k
mobilenetv3_small_075.lamb_in1k
mobilenetv3_small_100.lamb_in1k
mobilevit_xs
mobilevit_xxs
mobilevitv2_100
mobilevitv2_150_384_in22ft1k
regnetz_d8
rexnet_100
swin_large_patch4_window12_384
tf_efficientnet_b0.ns_jft_in1k
tf_efficientnet_b3.ns_jft_in1k
tf_efficientnet_b4.ns_jft_in1k
tf_efficientnet_b5.ns_jft_in1k
tf_efficientnet_b6.ns_jft_in1k
tf_efficientnet_b7.ns_jft_in1k
tf_efficientnetv2_b0.in1k
tf_mobilenetv3_small_100.in1k
tinynet_e.in1k
vit_base_patch16_clip_224.laion2b_ft_in12k_in1k
vit_base_patch16_clip_384.laion2b_ft_in12k_in1k
vit_base_patch32_clip_224.laion2b_ft_in12k_in1k
vit_base_patch32_clip_384.laion2b_ft_in12k_in1k
volo_d1_384
volo_d2_384
volo_d3_448
volo_d4_448
xcit_nano_12_p8_384_dist
xcit_small_12_p8_384_dist
xcit_small_24_p8_384_dist
xcit_tiny_12_p8_384_dist
xcit_tiny_24_p8_384_dist
```

Validation set models

adv_inception_v3
bat_resnext26ts.ch_in1k
beit_base_patch16_224.in22k_ft_in22k
beit_base_patch16_224.in22k_ft_in22k_in1k
beit_base_patch16_384.in22k_ft_in22k_in1k
beit_large_patch16_224.in22k_ft_in22k
beit_large_patch16_224.in22k_ft_in22k_in1k
beit_large_patch16_384.in22k_ft_in22k_in1k
beit_large_patch16_512.in22k_ft_in22k_in1k
beitv2_base_patch16_224.in1k_ft_in22k
beitv2_base_patch16_224.in1k_ft_in22k_in1k
beitv2_large_patch16_224.in1k_ft_in22k
beitv2_large_patch16_224.in1k_ft_in22k_in1k
botnet26t_256
cait_m36_384
cait_m48_448
cait_s24_224
cait_s24_384
cait_s36_384
cait_xs24_384
cait_xxs24_224
cait_xxs24_384
cait_xxs36_224
cait_xxs36_384
coat_lite_mini
coat_lite_small
coat_lite_tiny
coat_mini
coat_tiny
coatnet_0_rw_224.sw_in1k
coatnet_1_rw_224.sw_in1k
coatnet_2_rw_224.sw_in12k
coatnet_2_rw_224.sw_in12k_ft_in1k
coatnet_3_rw_224.sw_in12k
coatnet_bn_0_rw_224.sw_in1k
coatnet_nano_rw_224.sw_in1k
coatnet_rmlp_1_rw2_224.sw_in12k
coatnet_rmlp_1_rw2_224.sw_in12k_ft_in1k
coatnet_rmlp_1_rw_224.sw_in1k
coatnet_rmlp_2_rw_224.sw_in12k
coatnet_rmlp_2_rw_224.sw_in12k_ft_in1k
coatnet_rmlp_2_rw_224.sw_in1k
coatnet_rmlp_2_rw_384.sw_in12k_ft_in1k
coatnet_rmlp_nano_rw_224.sw_in1k
coatnext_nano_rw_224.sw_in1k
convit_base
convit_small
convit_tiny
convmixer_1024_20_ks9_p14
convmixer_1536_20
convmixer_768_32
convnext_atto.d2_in1k
convnext_atto_ols.a2_in1k
convnext_base.clip_laion2b
convnext_base.clip_laion2b_augreg
convnext_base.clip_laion2b_augreg_ft_in12k
convnext_base.clip_laion2b_augreg_ft_in12k_in1k
convnext_base.clip_laion2b_augreg_ft_in12k_in1k_384
convnext_base.clip_laion2b_augreg_ft_in1k
convnext_base.clip_laiona
convnext_base.clip_laiona_320
convnext_base.clip_laiona_augreg_320
convnext_base.clip_laiona_augreg_ft_in1k_384
convnext_base.fb_in1k
convnext_base.fb_in22k
convnext_base.fb_in22k_ft_in1k
convnext_base.fb_in22k_ft_in1k_384
convnext_femto.d1_in1k
convnext_femto_ols.d1_in1k
convnext_large.fb_in1k
convnext_large.fb_in22k
convnext_large.fb_in22k_ft_in1k
convnext_large.fb_in22k_ft_in1k_384
convnext_large_mlp.clip_laion2b_augreg
convnext_large_mlp.clip_laion2b_augreg_ft_in12k_384
convnext_large_mlp.clip_laion2b_augreg_ft_in1k
convnext_large_mlp.clip_laion2b_augreg_ft_in1k_384
convnext_large_mlp.clip_laion2b_ft_320
convnext_large_mlp.clip_laion2b_ft_soup_320
convnext_large_mlp.clip_laion2b_soup_ft_in12k_320
convnext_large_mlp.clip_laion2b_soup_ft_in12k_384
convnext_large_mlp.clip_laion2b_soup_ft_in12k_in1k_320
convnext_large_mlp.clip_laion2b_soup_ft_in12k_in1k_384
convnext_nano.d1h_in1k
convnext_nano.in12k
convnext_nano.in12k_ft_in1k
convnext_nano_ols.d1h_in1k
convnext_pico.d1_in1k
convnext_pico_ols.d1_in1k
convnext_small.fb_in1k
convnext_small.fb_in22k
convnext_small.fb_in22k_ft_in1k
convnext_small.fb_in22k_ft_in1k_384
convnext_small.in12k
convnext_small.in12k_ft_in1k
convnext_small.in12k_ft_in1k_384
convnext_tiny.fb_in1k
convnext_tiny.fb_in22k
convnext_tiny.fb_in22k_ft_in1k
convnext_tiny.fb_in22k_ft_in1k_384

convnext_tiny.in12k
convnext_tiny.in12k_ft_in1k
convnext_tiny.in12k_ft_in1k_384
convnext_tiny_hnf.a2h_in1k
convnext_xlarge.fb_in22k
convnext_xlarge.fb_in22k_ft_in1k
convnext_xlarge.fb_in22k_ft_in1k_384
convnext_xxlarge.clip_laion2b_rewind
convnext_xxlarge.clip_laion2b_soup
convnext_xxlarge.clip_laion2b_soup_ft_in1k
convnextv2_atto.fcmae
convnextv2_atto.fcmae_ft_in1k
convnextv2_base.fcmae
convnextv2_base.fcmae_ft_in1k
convnextv2_base.fcmae_ft_in22k_in1k
convnextv2_base.fcmae_ft_in22k_in1k_384
convnextv2_femto.fcmae
convnextv2_femto.fcmae_ft_in1k
convnextv2_huge.fcmae
convnextv2_huge.fcmae_ft_in1k
convnextv2_huge.fcmae_ft_in22k_in1k_384
convnextv2_huge.fcmae_ft_in22k_in1k_512
convnextv2_large.fcmae
convnextv2_large.fcmae_ft_in1k
convnextv2_large.fcmae_ft_in22k_in1k
convnextv2_large.fcmae_ft_in22k_in1k_384
convnextv2_nano.fcmae
convnextv2_nano.fcmae_ft_in1k
convnextv2_nano.fcmae_ft_in22k_in1k
convnextv2_nano.fcmae_ft_in22k_in1k_384
convnextv2_pico.fcmae
convnextv2_pico.fcmae_ft_in1k
convnextv2_tiny.fcmae
convnextv2_tiny.fcmae_ft_in1k
convnextv2_tiny.fcmae_ft_in22k_in1k
convnextv2_tiny.fcmae_ft_in22k_in1k_384
crossvit_15_240
crossvit_15_dagger_240
crossvit_15_dagger_408
crossvit_18_240
crossvit_18_dagger_240
crossvit_18_dagger_408
crossvit_9_240
crossvit_9_dagger_240
crossvit_base_240
crossvit_small_240
crossvit_tiny_240
cs3darknet_focus_l
cs3darknet_focus_m
cs3darknet_l
cs3darknet_m
cs3darknet_x
cs3edgenet_x
cs3se_edgenet_x
cs3sedarknet_l
cs3sedarknet_x
cspdarknet53
cspresnet50
cspresnext50
darknet53
darknetaa53
davit_base.msft_in1k
davit_small.msft_in1k
davit_tiny.msft_in1k
deit3_base_patch16_224.fb_in1k
deit3_base_patch16_224.fb_in22k_ft_in1k
deit3_base_patch16_384.fb_in1k
deit3_base_patch16_384.fb_in22k_ft_in1k
deit3_huge_patch14_224.fb_in1k
deit3_huge_patch14_224.fb_in22k_ft_in1k
deit3_large_patch16_224.fb_in1k
deit3_large_patch16_224.fb_in22k_ft_in1k
deit3_large_patch16_384.fb_in1k
deit3_large_patch16_384.fb_in22k_ft_in1k
deit3_medium_patch16_224.fb_in1k
deit3_medium_patch16_224.fb_in22k_ft_in1k
deit3_small_patch16_224.fb_in1k
deit3_small_patch16_224.fb_in22k_ft_in1k
deit3_small_patch16_384.fb_in1k
deit3_small_patch16_384.fb_in22k_ft_in1k
deit_base_distilled_patch16_224.fb_in1k
deit_base_distilled_patch16_384.fb_in1k
deit_base_patch16_224.fb_in1k
deit_base_patch16_384.fb_in1k
deit_small_distilled_patch16_224.fb_in1k
deit_small_patch16_224.fb_in1k
deit_tiny_distilled_patch16_224.fb_in1k
deit_tiny_patch16_224.fb_in1k
densenet121
densenet161
densenet169
densenet201
densenetblur121d
dla102
dla102x
dla102x2
dla169
dla34
dla46_c
dla46x_c
dla60
dla60_res2net
dla60_res2next
dla60x
dla60x_c
dm_nfnet_f0.dm_in1k
dm_nfnet_f1.dm_in1k
dm_nfnet_f2.dm_in1k

dm_nfnet_f3.dm_in1k
dm_nfnet_f4.dm_in1k
dm_nfnet_f5.dm_in1k
dm_nfnet_f6.dm_in1k
dpn107
dpn131
dpn68
dpn68b
dpn92
dpn98
eca_botnext26ts_256
eca_halonext26ts
eca_nfnet_l0.ra2_in1k
eca_nfnet_l1.ra2_in1k
eca_nfnet_l2.ra3_in1k
eca_resnet33ts.ra2_in1k
eca_resnext26ts.ch_in1k
ecaresnet101d.miil_in1k
ecaresnet101d_pruned.miil_in1k
ecaresnet269d.ra2_in1k
ecaresnet26t.ra2_in1k
ecaresnet50d.miil_in1k
ecaresnet50d_pruned.miil_in1k
ecaresnet50t.a1_in1k
ecaresnet50t.a2_in1k
ecaresnet50t.a3_in1k
ecaresnet50t.ra2_in1k
ecaresnetlight.miil_in1k
edgenext_base
edgenext_small
edgenext_small_rw
edgenext_x_small
edgenext_xx_small
efficientformer_l1.snap_dist_in1k
efficientformer_l3.snap_dist_in1k
efficientformer_l7.snap_dist_in1k
efficientformerv2_l.snap_dist_in1k
efficientformerv2_s0.snap_dist_in1k
efficientformerv2_s1.snap_dist_in1k
efficientformerv2_s2.snap_dist_in1k
efficientnet_b0.ra_in1k
efficientnet_b1.ft_in1k
efficientnet_b1_pruned.in1k
efficientnet_b2.ra_in1k
efficientnet_b2_pruned.in1k
efficientnet_b3.ra2_in1k
efficientnet_b3_pruned.in1k
efficientnet_b4.ra2_in1k
efficientnet_b5.in12k
efficientnet_b5.in12k_ft_in1k
efficientnet_el.ra_in1k
efficientnet_el_pruned.in1k
efficientnet_em.ra2_in1k
efficientnet_es.ra_in1k
efficientnet_es_pruned.in1k
efficientnet_lite0.ra_in1k
efficientnetv2_rw_m.agc_in1k
efficientnetv2_rw_s.ra2_in1k
efficientnetv2_rw_t.ra2_in1k
ens_adv_inception_resnet_v2
ese_vovnet19b_dw
ese_vovnet39b
eva02_base_patch14_224.mim_in22k
eva02_base_patch14_448.mim_in22k_ft_in1k
eva02_base_patch14_448.mim_in22k_ft_in22k
eva02_base_patch14_448.mim_in22k_ft_in22k_in1k
eva02_large_patch14_224.mim_in22k
eva02_large_patch14_224.mim_m38m
eva02_large_patch14_448.mim_in22k_ft_in1k
eva02_large_patch14_448.mim_in22k_ft_in22k
eva02_large_patch14_448.mim_in22k_ft_in22k_in1k
eva02_large_patch14_448.mim_m38m_ft_in1k
eva02_large_patch14_448.mim_m38m_ft_in22k
eva02_large_patch14_448.mim_m38m_ft_in22k_in1k
eva02_small_patch14_224.mim_in22k
eva02_small_patch14_336.mim_in22k_ft_in1k
eva02_tiny_patch14_224.mim_in22k
eva02_tiny_patch14_336.mim_in22k_ft_in1k
eva_giant_patch14_224.clip_ft_in1k
eva_giant_patch14_336.clip_ft_in1k
eva_giant_patch14_336.m30m_ft_in22k_in1k
eva_giant_patch14_560.m30m_ft_in22k_in1k
eva_large_patch14_196.in22k_ft_in1k
eva_large_patch14_196.in22k_ft_in22k_in1k
eva_large_patch14_336.in22k_ft_in1k
eva_large_patch14_336.in22k_ft_in22k_in1k
fbnetc_100.rmsp_in1k
fbnetv3_b.ra2_in1k
fbnetv3_d.ra2_in1k
fbnetv3_g.ra2_in1k
flexivit_base.1000ep_in21k
flexivit_base.1200ep_in1k
flexivit_base.300ep_in1k
flexivit_base.300ep_in21k
flexivit_base.600ep_in1k
flexivit_base.patch16_in21k
flexivit_base.patch30_in21k
flexivit_large.1200ep_in1k
flexivit_large.300ep_in1k
flexivit_large.600ep_in1k
flexivit_small.1200ep_in1k
flexivit_small.300ep_in1k
flexivit_small.600ep_in1k
focalnet_base_lrf.ms_in1k
focalnet_base_srf.ms_in1k
focalnet_huge_fl3.ms_in22k
focalnet_huge_fl4.ms_in22k
focalnet_large_fl3.ms_in22k

focalnet_large_fl4.ms_in22k
focalnet_small_lrf.ms_in1k
focalnet_small_srf.ms_in1k
focalnet_tiny_lrf.ms_in1k
focalnet_tiny_srf.ms_in1k
focalnet_xlarge_fl3.ms_in22k
focalnet_xlarge_fl4.ms_in22k
gc_efficientnetv2_rw_t.agc_in1k
gcresnet33ts.ra2_in1k
gcresnet50t.ra2_in1k
gcresnext26ts.ch_in1k
gcresnext50ts.ch_in1k
gcvit_base
gcvit_small
gcvit_tiny
gcvit_xtiny
gcvit_xxtiny
gernet_l.idstcv_in1k
gernet_m.idstcv_in1k
gernet_s.idstcv_in1k
ghostnet_100
gluon_inception_v3
gluon_xception65
gmixer_24_224.ra3_in1k
gmlp_s16_224.ra3_in1k
halo2botnet50ts_256
halonet26t
halonet50ts
haloregnetz_b
hardcorenas_a
hardcorenas_b
hardcorenas_c
hardcorenas_d
hardcorenas_e
hardcorenas_f
hrnet_w18
hrnet_w18_small
hrnet_w18_small_v2
hrnet_w30
hrnet_w32
hrnet_w40
hrnet_w44
hrnet_w48
hrnet_w64
inception_resnet_v2
inception_v3
inception_v4
jx_nest_base
jx_nest_small
jx_nest_tiny
lambda_resnet26rpt_256
lambda_resnet26t
lambda_resnet50ts
lamhalobotnet50ts_256
lcnet_050.ra2_in1k
lcnet_075.ra2_in1k
lcnet_100.ra2_in1k
legacy_senet154
legacy_seresnet101
legacy_seresnet152
legacy_seresnet18
legacy_seresnet34
legacy_seresnet50
legacy_seresnext101_32x4d
legacy_seresnext26_32x4d
legacy_seresnext50_32x4d
levit_128.fb_dist_in1k
levit_128s.fb_dist_in1k
levit_128s
levit_192.fb_dist_in1k
levit_256.fb_dist_in1k
levit_384.fb_dist_in1k
levit_conv_128.fb_dist_in1k
levit_conv_128s.fb_dist_in1k
levit_conv_192.fb_dist_in1k
levit_conv_256.fb_dist_in1k
levit_conv_384.fb_dist_in1k
maxvit_base_tf_224.in1k
maxvit_base_tf_384.in1k
maxvit_base_tf_384.in21k_ft_in1k
maxvit_base_tf_512.in1k
maxvit_base_tf_512.in21k_ft_in1k
maxvit_large_tf_224.in1k
maxvit_large_tf_384.in1k
maxvit_large_tf_384.in21k_ft_in1k
maxvit_large_tf_512.in1k
maxvit_large_tf_512.in21k_ft_in1k
maxvit_nano_rw_256.sw_in1k
maxvit_rmlp_base_rw_224.sw_in12k
maxvit_rmlp_base_rw_224.sw_in12k_ft_in1k
maxvit_rmlp_base_rw_384.sw_in12k_ft_in1k
maxvit_rmlp_nano_rw_256.sw_in1k
maxvit_rmlp_pico_rw_256.sw_in1k
maxvit_rmlp_small_rw_224.sw_in1k
maxvit_rmlp_tiny_rw_256.sw_in1k
maxvit_small_tf_224.in1k
maxvit_small_tf_384.in1k
maxvit_small_tf_512.in1k
maxvit_tiny_rw_224.sw_in1k
maxvit_tiny_tf_224.in1k
maxvit_tiny_tf_384.in1k
maxvit_tiny_tf_512.in1k
maxvit_xlarge_tf_384.in21k_ft_in1k
maxvit_xlarge_tf_512.in21k_ft_in1k
maxxvit_rmlp_nano_rw_256.sw_in1k
maxxvit_rmlp_small_rw_256.sw_in1k
maxxvitv2_nano_rw_256.sw_in1k
maxxvitv2_rmlp_base_rw_224.sw_in12k

maxxvitv2_rmlp_base_rw_224.sw_in12k_ft_in1k
maxxvitv2_rmlp_base_rw_384.sw_in12k_ft_in1k
mixer_b16_224.goog_in21k
mixer_b16_224.goog_in21k_ft_in1k
mixer_b16_224.miil_in21k
mixer_b16_224.miil_in21k_ft_in1k
mixer_l16_224.goog_in21k
mixer_l16_224.goog_in21k_ft_in1k
mixnet_l.ft_in1k
mixnet_m.ft_in1k
mixnet_s.ft_in1k
mixnet_xl.ra_in1k
mnasnet_100.rmsp_in1k
mnasnet_small.lamb_in1k
mobilenetv2_050.lamb_in1k
mobilenetv2_100.ra_in1k
mobilenetv2_110d.ra_in1k
mobilenetv2_120d.ra_in1k
mobilenetv2_140.ra_in1k
mobilenetv3_large_100.miil_in21k
mobilenetv3_large_100.miil_in21k_ft_in1k
mobilenetv3_large_100.ra_in1k
mobilenetv3_rw.rmsp_in1k
mobilenetv3_small_050.lamb_in1k
mobilenetv3_small_075.lamb_in1k
mobilenetv3_small_100.lamb_in1k
mobilevit_s
mobilevit_xs
mobilevit_xxs
mobilevitv2_050
mobilevitv2_075
mobilevitv2_100
mobilevitv2_125
mobilevitv2_150
mobilevitv2_150_384_in22ft1k
mobilevitv2_150_in22ft1k
mobilevitv2_175
mobilevitv2_175_384_in22ft1k
mobilevitv2_175_in22ft1k
mobilevitv2_200
mobilevitv2_200_384_in22ft1k
mobilevitv2_200_in22ft1k
mvitv2_base
mvitv2_large
mvitv2_small
mvitv2_tiny
nasnetalarge
nf_regnet_b1.ra2_in1k
nf_resnet50.ra2_in1k
nfnet_l0.ra2_in1k
pit_b_224
pit_b_distilled_224
pit_s_224
pit_s_distilled_224
pit_ti_224
pit_ti_distilled_224
pit_xs_224
pit_xs_distilled_224
pnasnet5large
poolformer_m36
poolformer_m48
poolformer_s12
poolformer_s24
poolformer_s36
pvt_v2_b0
pvt_v2_b1
pvt_v2_b2
pvt_v2_b2_li
pvt_v2_b3
pvt_v2_b4
pvt_v2_b5
regnetv_040.ra3_in1k
regnetv_064.ra3_in1k
regnetx_002.pycls_in1k
regnetx_004.pycls_in1k
regnetx_004_tv.tv2_in1k
regnetx_006.pycls_in1k
regnetx_008.pycls_in1k
regnetx_008.tv2_in1k
regnetx_016.pycls_in1k
regnetx_016.tv2_in1k
regnetx_032.pycls_in1k
regnetx_032.tv2_in1k
regnetx_040.pycls_in1k
regnetx_064.pycls_in1k
regnetx_080.pycls_in1k
regnetx_080.tv2_in1k
regnetx_120.pycls_in1k
regnetx_160.pycls_in1k
regnetx_160.tv2_in1k
regnetx_320.pycls_in1k
regnetx_320.tv2_in1k
regnety_002.pycls_in1k
regnety_004.pycls_in1k
regnety_004.tv2_in1k
regnety_006.pycls_in1k
regnety_008.pycls_in1k
regnety_008_tv.tv2_in1k
regnety_016.pycls_in1k
regnety_016.tv2_in1k
regnety_032.pycls_in1k
regnety_032.ra_in1k
regnety_032.tv2_in1k
regnety_040.pycls_in1k
regnety_040.ra3_in1k
regnety_064.pycls_in1k
regnety_064.ra3_in1k
regnety_080.pycls_in1k

regnety_080.ra3_in1k
regnety_080_tv.tv2_in1k
regnety_120.pycls_in1k
regnety_120.sw_in12k
regnety_120.sw_in12k_ft_in1k
regnety_1280.seer
regnety_1280.seer_ft_in1k
regnety_1280.swag_ft_in1k
regnety_1280.swag_lc_in1k
regnety_160.deit_in1k
regnety_160.lion_in12k_ft_in1k
regnety_160.pycls_in1k
regnety_160.sw_in12k
regnety_160.sw_in12k_ft_in1k
regnety_160.swag_ft_in1k
regnety_160.swag_lc_in1k
regnety_160.tv2_in1k
regnety_2560.seer_ft_in1k
regnety_320.pycls_in1k
regnety_320.seer
regnety_320.seer_ft_in1k
regnety_320.swag_ft_in1k
regnety_320.swag_lc_in1k
regnety_320.tv2_in1k
regnety_640.seer
regnety_640.seer_ft_in1k
regnetz_040.ra3_in1k
regnetz_040_h.ra3_in1k
regnetz_b16.ra3_in1k
regnetz_c16.ra3_in1k
regnetz_c16_evos.ch_in1k
regnetz_d32.ra3_in1k
regnetz_d8
regnetz_d8.ra3_in1k
regnetz_d8_evos.ch_in1k
regnetz_e8.ra3_in1k
repvgg_a2.rvgg_in1k
repvgg_b0.rvgg_in1k
repvgg_b1.rvgg_in1k
repvgg_b1g4.rvgg_in1k
repvgg_b2.rvgg_in1k
repvgg_b2g4.rvgg_in1k
repvgg_b3.rvgg_in1k
repvgg_b3g4.rvgg_in1k
res2net101_26w_4s
res2net50_14w_8s
res2net50_26w_4s
res2net50_26w_6s
res2net50_26w_8s
res2net50_48w_2s
res2next50
resmlp_12_224.fb_dino
resmlp_12_224.fb_distilled_in1k
resmlp_12_224.fb_in1k
resmlp_24_224.fb_dino
resmlp_24_224.fb_distilled_in1k
resmlp_24_224.fb_in1k
resmlp_36_224.fb_distilled_in1k
resmlp_36_224.fb_in1k
resmlp_big_24_224.fb_distilled_in1k
resmlp_big_24_224.fb_in1k
resmlp_big_24_224.fb_in22k_ft_in1k
resnest101e
resnest14d
resnest200e
resnest269e
resnest26d
resnest50d
resnest50d_1s4x24d
resnest50d_4s2x40d
resnet101.a1_in1k
resnet101.a1h_in1k
resnet101.a2_in1k
resnet101.a3_in1k
resnet101.gluon_in1k
resnet101.tv2_in1k
resnet101.tv_in1k
resnet101c.gluon_in1k
resnet101d.gluon_in1k
resnet101d.ra2_in1k
resnet101s.gluon_in1k
resnet10t.c3_in1k
resnet14t.c3_in1k
resnet152.a1_in1k
resnet152.a1h_in1k
resnet152.a2_in1k
resnet152.a3_in1k
resnet152.gluon_in1k
resnet152.tv2_in1k
resnet152.tv_in1k
resnet152c.gluon_in1k
resnet152d.gluon_in1k
resnet152d.ra2_in1k
resnet152s.gluon_in1k
resnet18.a1_in1k
resnet18.a2_in1k
resnet18.a3_in1k
resnet18.fb_ssl_yfcc100m_ft_in1k
resnet18.fb_swsl_ig1b_ft_in1k
resnet18.gluon_in1k
resnet18.tv_in1k
resnet18d.ra2_in1k
resnet200d.ra2_in1k
resnet26.bt_in1k
resnet26d.bt_in1k
resnet26t.ra2_in1k
resnet32ts.ra2_in1k
resnet33ts.ra2_in1k

resnet34.a1_in1k
resnet34.a2_in1k
resnet34.a3_in1k
resnet34.bt_in1k
resnet34.gluon_in1k
resnet34.tv_in1k
resnet34d.ra2_in1k
resnet50.a1_in1k
resnet50.a1h_in1k
resnet50.a2_in1k
resnet50.a3_in1k
resnet50.am_in1k
resnet50.b1k_in1k
resnet50.b2k_in1k
resnet50.bt_in1k
resnet50.c1_in1k
resnet50.c2_in1k
resnet50.d_in1k
resnet50.fb_ssl_yfcc100m_ft_in1k
resnet50.fb_swsl_ig1b_ft_in1k
resnet50.gluon_in1k
resnet50.ra_in1k
resnet50.ram_in1k
resnet50.tv2_in1k
resnet50.tv_in1k
resnet50_gn.a1h_in1k
resnet50c.gluon_in1k
resnet50d.a1_in1k
resnet50d.a2_in1k
resnet50d.a3_in1k
resnet50d.gluon_in1k
resnet50d.ra2_in1k
resnet50s.gluon_in1k
resnet51q.ra2_in1k
resnet61q.ra2_in1k
resnetaa101d.sw_in12k
resnetaa101d.sw_in12k_ft_in1k
resnetaa50.a1h_in1k
resnetaa50d.d_in12k
resnetaa50d.sw_in12k
resnetaa50d.sw_in12k_ft_in1k
resnetblur50.bt_in1k
resnetrs101.tf_in1k
resnetrs152.tf_in1k
resnetrs200.tf_in1k
resnetrs270.tf_in1k
resnetrs350.tf_in1k
resnetrs420.tf_in1k
resnetrs50.tf_in1k
resnetv2_101.a1h_in1k
resnetv2_101x1_bit.goog_in21k
resnetv2_101x1_bit.goog_in21k_ft_in1k
resnetv2_101x3_bit.goog_in21k
resnetv2_101x3_bit.goog_in21k_ft_in1k
resnetv2_152x2_bit.goog_in21k
resnetv2_152x2_bit.goog_in21k_ft_in1k
resnetv2_152x2_bit.goog_teacher_in21k_ft_in1k
resnetv2_152x2_bit.goog_teacher_in21k_ft_in1k_384
resnetv2_152x4_bit.goog_in21k
resnetv2_152x4_bit.goog_in21k_ft_in1k
resnetv2_50.a1h_in1k
resnetv2_50d_evos.ah_in1k
resnetv2_50d_gn.ah_in1k
resnetv2_50x1_bit.goog_distilled_in1k
resnetv2_50x1_bit.goog_in21k
resnetv2_50x1_bit.goog_in21k_ft_in1k
resnetv2_50x3_bit.goog_in21k
resnetv2_50x3_bit.goog_in21k_ft_in1k
resnext101_32x16d.fb_ssl_yfcc100m_ft_in1k
resnext101_32x16d.fb_swsl_ig1b_ft_in1k
resnext101_32x16d.fb_wsl_ig1b_ft_in1k
resnext101_32x32d.fb_wsl_ig1b_ft_in1k
resnext101_32x4d.fb_ssl_yfcc100m_ft_in1k
resnext101_32x4d.fb_swsl_ig1b_ft_in1k
resnext101_32x4d.gluon_in1k
resnext101_32x8d.fb_ssl_yfcc100m_ft_in1k
resnext101_32x8d.fb_swsl_ig1b_ft_in1k
resnext101_32x8d.fb_wsl_ig1b_ft_in1k
resnext101_32x8d.tv2_in1k
resnext101_32x8d.tv_in1k
resnext101_64x4d.c1_in1k
resnext101_64x4d.gluon_in1k
resnext101_64x4d.tv_in1k
resnext26ts.ra2_in1k
resnext50_32x4d.a1_in1k
resnext50_32x4d.a1h_in1k
resnext50_32x4d.a2_in1k
resnext50_32x4d.a3_in1k
resnext50_32x4d.fb_ssl_yfcc100m_ft_in1k
resnext50_32x4d.fb_swsl_ig1b_ft_in1k
resnext50_32x4d.gluon_in1k
resnext50_32x4d.ra_in1k
resnext50_32x4d.tv2_in1k
resnext50_32x4d.tv_in1k
resnext50d_32x4d.bt_in1k
rexnet_100.nav_in1k
rexnet_100
rexnet_130.nav_in1k
rexnet_150.nav_in1k
rexnet_200.nav_in1k
rexnet_300.nav_in1k
rexnetr_200.sw_in12k
rexnetr_200.sw_in12k_ft_in1k
rexnetr_300.sw_in12k
rexnetr_300.sw_in12k_ft_in1k
sebotnet33ts_256
sehalonet33ts
selecsls42b

selecsls60
selecsls60b
semnasnet_075.rmsp_in1k
semnasnet_100.rmsp_in1k
senet154.gluon_in1k
sequencer2d_l
sequencer2d_m
sequencer2d_s
seresnet152d.ra2_in1k
seresnet33ts.ra2_in1k
seresnet50.a1_in1k
seresnet50.a2_in1k
seresnet50.a3_in1k
seresnet50.ra2_in1k
seresnext101_32x4d.gluon_in1k
seresnext101_32x8d.ah_in1k
seresnext101_64x4d.gluon_in1k
seresnext101d_32x8d.ah_in1k
seresnext26d_32x4d.bt_in1k
seresnext26t_32x4d.bt_in1k
seresnext26ts.ch_in1k
seresnext50_32x4d.gluon_in1k
seresnext50_32x4d.racm_in1k
seresnextaa101d_32x8d.ah_in1k
seresnextaa101d_32x8d.sw_in12k
seresnextaa101d_32x8d.sw_in12k_ft_in1k
seresnextaa101d_32x8d.sw_in12k_ft_in1k_288
skresnet18
skresnet34
skresnext50_32x4d
spnasnet_100.rmsp_in1k
swin_base_patch4_window12_384.ms_in1k
swin_base_patch4_window12_384.ms_in22k
swin_base_patch4_window12_384.ms_in22k_ft_in1k
swin_base_patch4_window7_224.ms_in1k
swin_base_patch4_window7_224.ms_in22k
swin_base_patch4_window7_224.ms_in22k_ft_in1k
swin_large_patch4_window12_384.ms_in22k
swin_large_patch4_window12_384.ms_in22k_ft_in1k
swin_large_patch4_window12_384
swin_large_patch4_window7_224.ms_in22k
swin_large_patch4_window7_224.ms_in22k_ft_in1k
swin_s3_base_224.ms_in1k
swin_s3_small_224.ms_in1k
swin_s3_tiny_224.ms_in1k
swin_small_patch4_window7_224.ms_in1k
swin_small_patch4_window7_224.ms_in22k
swin_small_patch4_window7_224.ms_in22k_ft_in1k
swin_tiny_patch4_window7_224.ms_in1k
swin_tiny_patch4_window7_224.ms_in22k
swin_tiny_patch4_window7_224.ms_in22k_ft_in1k
swinv2_base_window12_192.ms_in22k
swinv2_base_window12to16_192to256.ms_in22k_ft_in1k
swinv2_base_window12to24_192to384.ms_in22k_ft_in1k
swinv2_base_window16_256.ms_in1k
swinv2_base_window8_256.ms_in1k
swinv2_cr_small_224.sw_in1k
swinv2_cr_small_ns_224.sw_in1k
swinv2_cr_tiny_ns_224.sw_in1k
swinv2_large_window12_192.ms_in22k
swinv2_large_window12to16_192to256.ms_in22k_ft_in1k
swinv2_large_window12to24_192to384.ms_in22k_ft_in1k
swinv2_small_window16_256.ms_in1k
swinv2_small_window8_256.ms_in1k
swinv2_tiny_window16_256.ms_in1k
swinv2_tiny_window8_256.ms_in1k
tf_efficientnet_b0.aa_in1k
tf_efficientnet_b0.ap_in1k
tf_efficientnet_b0.ns_jft_in1k
tf_efficientnet_b1.aa_in1k
tf_efficientnet_b1.ap_in1k
tf_efficientnet_b1.ns_jft_in1k
tf_efficientnet_b2.aa_in1k
tf_efficientnet_b2.ap_in1k
tf_efficientnet_b2.ns_jft_in1k
tf_efficientnet_b3.aa_in1k
tf_efficientnet_b3.ap_in1k
tf_efficientnet_b3.ns_jft_in1k
tf_efficientnet_b4.aa_in1k
tf_efficientnet_b4.ap_in1k
tf_efficientnet_b4.ns_jft_in1k
tf_efficientnet_b5.ap_in1k
tf_efficientnet_b5.ns_jft_in1k
tf_efficientnet_b5.ra_in1k
tf_efficientnet_b6.aa_in1k
tf_efficientnet_b6.ap_in1k
tf_efficientnet_b6.ns_jft_in1k
tf_efficientnet_b7.ap_in1k
tf_efficientnet_b7.ns_jft_in1k
tf_efficientnet_b7.ra_in1k
tf_efficientnet_b8.ap_in1k
tf_efficientnet_b8.ra_in1k
tf_efficientnet_cc_b0_4e.in1k
tf_efficientnet_cc_b0_8e.in1k
tf_efficientnet_cc_b1_8e.in1k
tf_efficientnet_el.in1k
tf_efficientnet_em.in1k
tf_efficientnet_es.in1k
tf_efficientnet_lite0.in1k
tf_efficientnet_lite1.in1k
tf_efficientnet_lite2.in1k
tf_efficientnet_lite3.in1k
tf_efficientnet_lite4.in1k
tf_efficientnetv2_b0.in1k
tf_efficientnetv2_b1.in1k
tf_efficientnetv2_b2.in1k
tf_efficientnetv2_b3.in1k
tf_efficientnetv2_b3.in21k

tf_efficientnetv2_b3.in21k_ft_in1k
tf_efficientnetv2_l.in1k
tf_efficientnetv2_l.in21k
tf_efficientnetv2_l.in21k_ft_in1k
tf_efficientnetv2_m.in1k
tf_efficientnetv2_m.in21k
tf_efficientnetv2_m.in21k_ft_in1k
tf_efficientnetv2_s.in1k
tf_efficientnetv2_s.in21k
tf_efficientnetv2_s.in21k_ft_in1k
tf_efficientnetv2_xl.in21k
tf_efficientnetv2_xl.in21k_ft_in1k
tf_inception_v3
tf_mixnet_l.in1k
tf_mixnet_m.in1k
tf_mixnet_s.in1k
tf_mobilenetv3_large_075.in1k
tf_mobilenetv3_large_100.in1k
tf_mobilenetv3_large_minimal_100.in1k
tf_mobilenetv3_small_075.in1k
tf_mobilenetv3_small_100.in1k
tf_mobilenetv3_small_minimal_100.in1k
tinynet_a.in1k
tinynet_b.in1k
tinynet_c.in1k
tinynet_d.in1k
tinynet_e.in1k
tnt_s_patch16_224
tv_densenet121
twins_pcpvt_base
twins_pcpvt_large
twins_pcpvt_small
twins_svt_base
twins_svt_large
twins_svt_small
vgg11
vgg11_bn
vgg13
vgg13_bn
vgg16
vgg16_bn
vgg19
vgg19_bn
visformer_small
vit_base_patch16_224.augreg2_in21k_ft_in1k
vit_base_patch16_224.augreg_in1k
vit_base_patch16_224.augreg_in21k
vit_base_patch16_224.augreg_in21k_ft_in1k
vit_base_patch16_224.dino
vit_base_patch16_224.orig_in21k_ft_in1k
vit_base_patch16_224.sam
vit_base_patch16_224_miil.in21k
vit_base_patch16_224_miil.in21k_ft_in1k
vit_base_patch16_384.augreg_in1k
vit_base_patch16_384.augreg_in21k_ft_in1k
vit_base_patch16_384.orig_in21k_ft_in1k
vit_base_patch16_clip_224.laion2b
vit_base_patch16_clip_224.laion2b_ft_in12k
vit_base_patch16_clip_224.laion2b_ft_in12k_in1k
vit_base_patch16_clip_224.laion2b_ft_in1k
vit_base_patch16_clip_224.openai
vit_base_patch16_clip_224.openai_ft_in12k
vit_base_patch16_clip_224.openai_ft_in12k_in1k
vit_base_patch16_clip_224.openai_ft_in1k
vit_base_patch16_clip_384.laion2b_ft_in12k_in1k
vit_base_patch16_clip_384.laion2b_ft_in1k
vit_base_patch16_clip_384.openai_ft_in12k_in1k
vit_base_patch16_clip_384.openai_ft_in1k
vit_base_patch16_rpn_224.in1k
vit_base_patch32_224.augreg_in1k
vit_base_patch32_224.augreg_in21k
vit_base_patch32_224.augreg_in21k_ft_in1k
vit_base_patch32_224.sam
vit_base_patch32_384.augreg_in1k
vit_base_patch32_384.augreg_in21k_ft_in1k
vit_base_patch32_clip_224.laion2b
vit_base_patch32_clip_224.laion2b_ft_in12k_in1k
vit_base_patch32_clip_224.laion2b_ft_in1k
vit_base_patch32_clip_224.openai
vit_base_patch32_clip_224.openai_ft_in1k
vit_base_patch32_clip_384.laion2b_ft_in12k_in1k
vit_base_patch32_clip_384.openai_ft_in12k_in1k
vit_base_patch32_clip_448.laion2b_ft_in12k_in1k
vit_base_patch8_224.augreg2_in21k_ft_in1k
vit_base_patch8_224.augreg_in21k
vit_base_patch8_224.augreg_in21k_ft_in1k
vit_base_patch8_224.dino
vit_base_r50_s16_224.orig_in21k
vit_base_r50_s16_384.orig_in21k_ft_in1k
vit_giant_patch14_clip_224.laion2b
vit_gigantic_patch14_clip_224.laion2b
vit_huge_patch14_224.orig_in21k
vit_huge_patch14_clip_224.laion2b
vit_huge_patch14_clip_224.laion2b_ft_in12k
vit_huge_patch14_clip_224.laion2b_ft_in12k_in1k
vit_huge_patch14_clip_224.laion2b_ft_in1k
vit_huge_patch14_clip_336.laion2b_ft_in12k_in1k
vit_large_patch14_clip_224.laion2b
vit_large_patch14_clip_224.laion2b_ft_in12k
vit_large_patch14_clip_224.laion2b_ft_in12k_in1k
vit_large_patch14_clip_224.laion2b_ft_in1k
vit_large_patch14_clip_224.openai
vit_large_patch14_clip_224.openai_ft_in12k
vit_large_patch14_clip_224.openai_ft_in12k_in1k
vit_large_patch14_clip_224.openai_ft_in1k
vit_large_patch14_clip_336.laion2b_ft_in12k_in1k
vit_large_patch14_clip_336.laion2b_ft_in1k
vit_large_patch14_clip_336.openai_ft_in12k_in1k

vit_large_patch16_224.augreg_in21k
vit_large_patch16_384.augreg_in21k_ft_in1k
vit_large_patch32_384.orig_in21k_ft_in1k
vit_large_r50_s32_224.augreg_in21k_ft_in1k
vit_medium_patch16_gap_240.in12k
vit_medium_patch16_gap_384.in12k_ft_in1k
vit_relpos_base_patch16_clsgap_224.sw_in1k
vit_relpos_medium_patch16_224.sw_in1k
vit_relpos_medium_patch16_rpn_224.sw_in1k
vit_small_patch16_224.augreg_in1k
vit_small_patch16_224.augreg_in21k_ft_in1k
vit_small_patch16_384.augreg_in1k
vit_small_patch32_224.augreg_in21k
vit_small_patch32_384.augreg_in21k_ft_in1k
vit_small_r26_s32_224.augreg_in21k
vit_small_r26_s32_384.augreg_in21k_ft_in1k
vit_srelpos_small_patch16_224.sw_in1k
vit_tiny_patch16_224.augreg_in21k_ft_in1k
vit_tiny_r_s16_p8_224.augreg_in21k
vit_tiny_r_s16_p8_384.augreg_in21k_ft_in1k
volo_d1_384
volo_d2_384
volo_d3_448
volo_d4_448
volo_d5_448
wide_resnet101_2.tv2_in1k
wide_resnet50_2.racm_in1k
wide_resnet50_2.tv_in1k
xception41
xception65
xception71
xcit_large_24_p16_224_dist
xcit_large_24_p8_224
xcit_large_24_p8_384_dist
xcit_medium_24_p16_224_dist
xcit_medium_24_p8_224
xcit_medium_24_p8_384_dist
xcit_nano_12_p16_224_dist
xcit_nano_12_p8_224
xcit_nano_12_p8_384_dist
xcit_small_12_p16_224_dist
xcit_small_12_p8_224
xcit_small_12_p8_384_dist
xcit_small_24_p16_224_dist
xcit_small_24_p8_224
xcit_small_24_p8_384_dist
xcit_tiny_12_p16_224_dist
xcit_tiny_12_p8_224
xcit_tiny_12_p8_384_dist
xcit_tiny_24_p16_224_dist
xcit_tiny_24_p8_224
xcit_tiny_24_p8_384_dist

vit_large_patch16_224.augreg_in21k_ft_in1k
vit_large_patch32_224.orig_in21k
vit_large_r50_s32_224.augreg_in21k
vit_large_r50_s32_384.augreg_in21k_ft_in1k
vit_medium_patch16_gap_256.in12k_ft_in1k
vit_relpos_base_patch16_224.sw_in1k
vit_relpos_base_patch32_plus_rpn_256.sw_in1k
vit_relpos_medium_patch16_cls_224.sw_in1k
vit_relpos_small_patch16_224.sw_in1k
vit_small_patch16_224.augreg_in21k
vit_small_patch16_224.dino
vit_small_patch16_384.augreg_in21k_ft_in1k
vit_small_patch32_224.augreg_in21k_ft_in1k
vit_small_patch8_224.dino
vit_small_r26_s32_224.augreg_in21k_ft_in1k
vit_srelpos_medium_patch16_224.sw_in1k
vit_tiny_patch16_224.augreg_in21k
vit_tiny_patch16_384.augreg_in21k_ft_in1k
vit_tiny_r_s16_p8_224.augreg_in21k_ft_in1k
volo_d1_224
volo_d2_224
volo_d3_224
volo_d4_224
volo_d5_224
volo_d5_512
wide_resnet101_2.tv_in1k
wide_resnet50_2.tv2_in1k
xception
xception41p
xception65p
xcit_large_24_p16_224
xcit_large_24_p16_384_dist
xcit_large_24_p8_224_dist
xcit_medium_24_p16_224
xcit_medium_24_p16_384_dist
xcit_medium_24_p8_224_dist
xcit_nano_12_p16_224
xcit_nano_12_p16_384_dist
xcit_nano_12_p8_224_dist
xcit_small_12_p16_224
xcit_small_12_p16_384_dist
xcit_small_12_p8_224_dist
xcit_small_24_p16_224
xcit_small_24_p16_384_dist
xcit_small_24_p8_224_dist
xcit_tiny_12_p16_224
xcit_tiny_12_p16_384_dist
xcit_tiny_12_p8_224_dist
xcit_tiny_24_p16_224
xcit_tiny_24_p16_384_dist
xcit_tiny_24_p8_224_dist