Clustering of Unseen Datasets with Self-Supervised Encoders

Joakim Bruslund Haurum
Assistant Professor
Center for Software Technology - SDU Vejle

SDU

An Empirical Study into Clustering of Unseen Datasets with Self-Supervised Encoders

Scott C. Lowe[*1], Joakim Bruslund Haurum[*2,3], Sageev Oore[†1,4], Thomas B. Moeslund[†2,3], and Graham W. Taylor[†1,5]

[1]Vector Institute, Canada, [2]Aalborg University, Denmark, [3]Pioneer Centre for AI, Denmark, [4]Dalhousie University, Canada, [5]University of Guelph, Canada

scott.lowe@vectorinstitute.ai {joha,tbm}@create.aau.dk
sageev@dal.ca gwtaylor@uoguelph.ca

http://scottclowe.com/zs-ssl-clustering/

Abstract

Can pretrained models generalize to new datasets without any retraining? We deploy pretrained image models on datasets they were not trained for, and investigate whether their embeddings form meaningful clusters. Our suite of benchmarking experiments use encoders pretrained solely on ImageNet-1k with either supervised or self-supervised training techniques, deployed on image datasets that were not seen during training, and clustered with conventional clustering algorithms. This evaluation provides new insights into the embeddings of self-supervised models, which prioritize different features to supervised models. Supervised encoders typically offer more utility than SSL encoders within the training domain, and vice-versa far outside of it, however, fine-tuned encoders demonstrate the opposite trend. Clustering provides a way to evaluate the utility of self-supervised learned representations orthogonal to existing methods such as kNN. Additionally, we find the silhouette score when measured in a UMAP-reduced space is highly correlated with clustering performance, and can therefore be used as a proxy for clustering performance on data with no ground truth labels. Our code implementation is available at https://github.com/scottclowe/zs-ssl-clustering/.

1 Introduction

Self-supervised learning (SSL) has attracted great interest in recent years across almost every machine learning sub-field, due to the promise of being able to harness large quantities of unlabelled data and obtaining generic feature embeddings useful for a variety of downstream tasks (Balestriero et al., 2023). This has, for example, led to the development of impressive large language models (Brown et al., 2020) and computer vision systems trained on 1 billion images (Goyal et al., 2021). However, while the embeddings from an SSL-trained encoder can perform well on downstream tasks after fine-tuning the network, there has been less investigation into the utility of the embeddings without fine-tuning. Prior work (Vaze et al., 2022; Zhou and Zhang, 2022) suggests SSL feature encoders generate embeddings suitable for clustering, but nonetheless adjust the feature encoders through fine-tuning. Yet, widespread interest in the application of large pretrained models on custom datasets, combined with prohibitive cost of compute, make this question important and increasingly urgent.

We find that to date there has been no investigation into whether SSL-trained feature encoders can serve as a foundation for clustering, yielding informative groupings of embeddings on real-world datasets that were totally unseen to the encoder during its training. Vaze et al. (2023) showed that features from SSL encoders are typically biased toward shape features and not color, texture, or count when clustered using K-Means. However, this was conducted using a synthetic dataset, where very specific object attributes could be disentangled. In contrast, in this work we perform a *zero-shot transfer-learning task*, evaluating the

*Joint first author. †Joint last author.

The Cost of Training Networks

Training Large Scale Networks

Training Large Scale Networks

→ Modern ”foundation” models can be prohibitevely expensive to train

Training Large Scale Networks

→ Modern ”foundation” models can be prohibitevely expensive to train

Model to Reproduce	GPU Type	GPU Power consumption	GPU-hours	PUE	Total power consumption	Carbon emitted (tCO_2eq)
DINOv2-g	A100-40GB	400W	22,016	1.1	9.7 MWh	3.7

DINOv2: Learning Robust Visual Features without Supervision, Oquab et al., 2024

Training Large Scale Networks

→ Modern ”foundation” models can be prohibitevely expensive to train

→ Not sustainable (or feasible) to retrain these models

Model to Reproduce	GPU Type	GPU Power consumption	GPU-hours	PUE	Total power consumption	Carbon emitted (tCO_2eq)
DINOv2-g	A100-40GB	400W	22,016	1.1	9.7 MWh	3.7

DINOv2: Learning Robust Visual Features without Supervision, Oquab et al., 2024

Training Large Scale Networks

- → Modern ”foundation” models can be prohibitevely expensive to train
- → Not sustainable (or feasible) to retrain these models
- → How can such models be efficiently reused?

Model to Reproduce	GPU Type	GPU Power consumption	GPU-hours	PUE	Total power consumption	Carbon emitted (tCO_2eq)
DINOv2-g	A100-40GB	400W	22,016	1.1	9.7 MWh	3.7

DINOv2: Learning Robust Visual Features without Supervision, Oquab et al., 2024

Evaluation through Probing

Evaluation through Probing

→ Probing is a way of converting a SSL pretained encoder into a classifiers

→ Typically done via kNN prbing or linear probing

Evaluation through Probing

→ Probing is a way of converting a SSL pretained encoder into a classifiers

→ Typically done via kNN prbing or linear probing

→ kNN probing = for each test point assign label based on k closest samples from a labeled dataset

→ Linear probing = train a Linear layer on top of the frozen backbone (ie logistic regression)

Evaluation through Probing

→ Probing is a way of converting a SSL pretained encoder into a classifiers

→ Typically done via kNN prbing or linear probing

→ kNN probing = for each test point assign label based on k closest samples from a labeled dataset

→ Linear probing = train a Linear layer on top of the frozen backbone (ie logistic regression)

→ But what if you don’t want to train anything or don’t have a labeled dataset?

→ Could clustering of features be a viable alternative?

Zero-Shot Clustering

The Main Idea

How well does features from (SSL)-pretrained networks cluster?

The Main Idea

How well does features from (SSL)-pretrained networks cluster?

→ We take already trained networks

→ Only fit a classical clustering method on a limited set of ”training” datasets

→ Transfer encoders and fitted clustering method to new datasets – *no adjustments made!*

Encoders

Encoders

→ We only consider networks trained on ImageNet-1K

→ Most commonly used backbones for SSL pretraining (at the time) are: ViT-B and ResNet-50

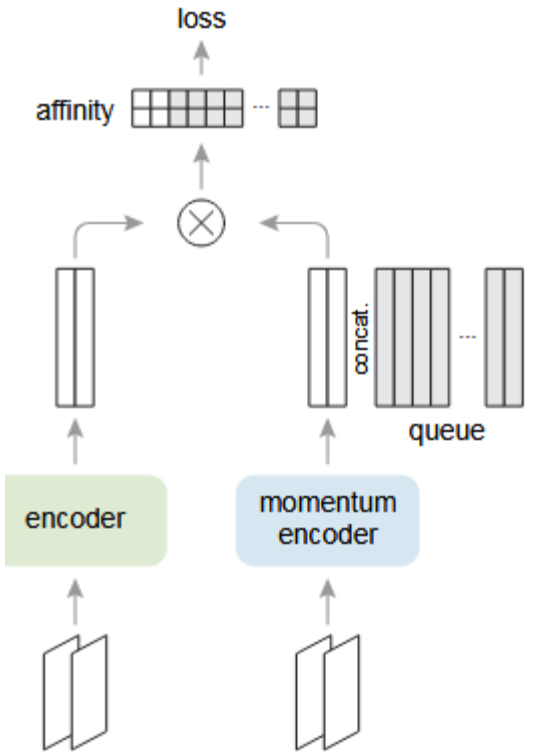
Encoders

→ Cross-Entropy Supervised

Encoders

→ Cross-Entropy Supervised

→ Contrastive Learning: MoCo-v3



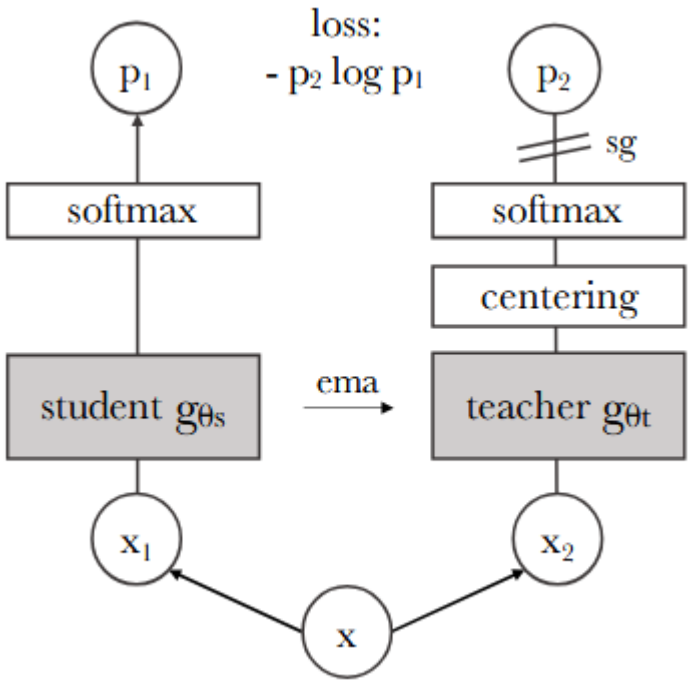
Improved Baselines with Momentum Contrastive Learning, Chen et al, 2020

Encoders

→ Cross-Entropy Supervised

→ Contrastive Learning: MoCo-v3

→ Self-Distillation: DINO



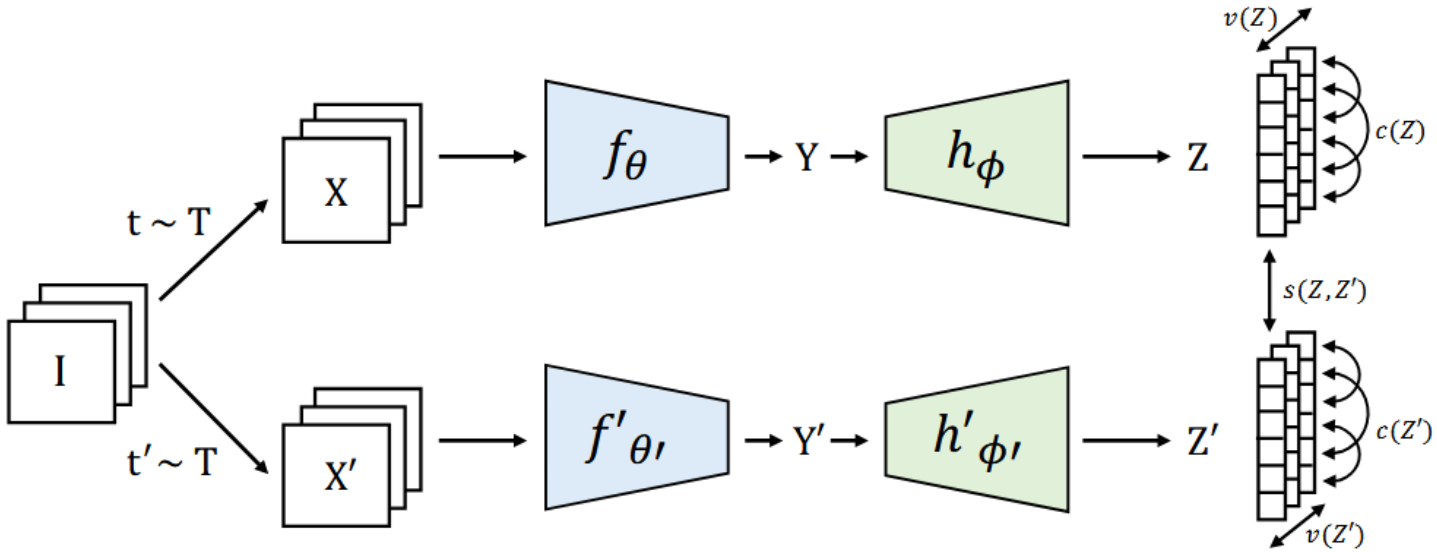
Encoders

→ Cross-Entropy Supervised

→ Contrastive Learning: MoCo-v3

→ Self-Distillation: DINO

→ Canonical Correlation Analysis: VICReg



VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning, Bardes et al., 2022

Encoders

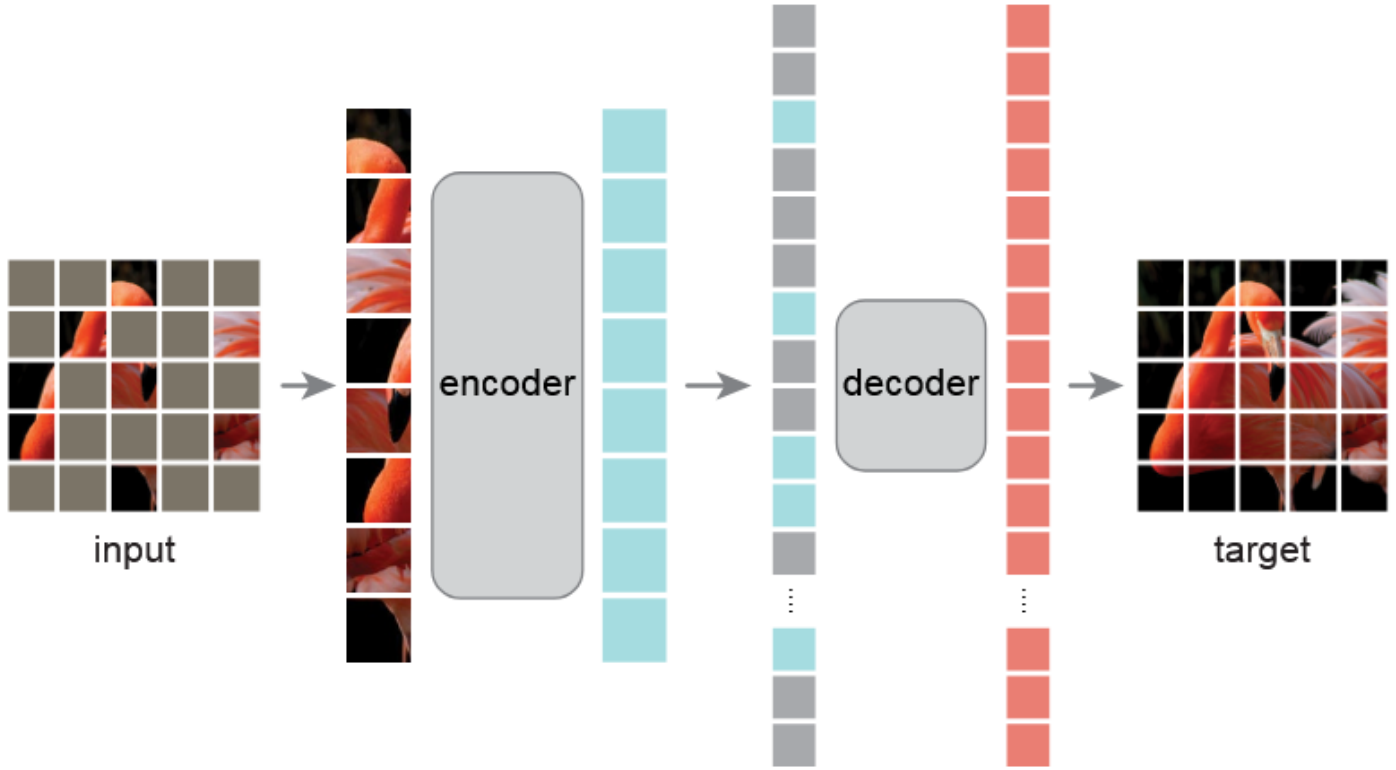
→ Cross-Entropy Supervised

→ Contrastive Learning: MoCo-v3

→ Self-Distillation: DINO

→ Canonical Correlation Analysis: VICReg

→ Masked Image Modelling: MAE



Masked Autoencoders Are Scalable Vision Learner, He et al., 2021

Clustering Methods

Clustering Methods

→ Partitioning-based: **K-Means**

Clustering Methods

→ Partitioning-based: **K-Means**

→ Hierarchical: **Agglomerative Clustering** (AC)
  → With and without known number of clusters

Clustering Methods

→ Partitioning-based: **K-Means**

→ Hierarchical: **Agglomerative Clustering** (AC)
  → With and without known number of clusters

→ Graph theoretical: **Affinity Propagation** (AP) & **Spectral**

Clustering Methods

→ Partitioning-based: **K-Means**

→ Hierarchical: **Agglomerative Clustering** (AC)
  → With and without known number of clusters

→ Graph theoretical: **Affinity Propagation** (AP) & **Spectral**

→ Density-based: **HDBSCAN**

Dimensionality Reduction

Dimensionality Reduction

→ Classic clustering method often works best with dimensionality reduction

Dimensionality Reduction

→ Classic clustering method often works best with dimensionality reduction

→ Principle Component Analysis (PCA)

→ Uniform Manifold Approximation and Projection (UMAP)

→ Pairwise Controlled Manifold Approximation Projection (PaCMAP)

Evaluation Metric

Evaluation Metric

→ We evaluate constructed clusters by comparing to Ground Truth labels

Evaluation Metric

→ We evaluate constructed clusters by comparing to Ground Truth labels

→ Normalized Mututal Information (NMI) is a common metric:

Evaluation Metric

→ We evaluate constructed clusters by comparing to Ground Truth labels

→ Normalized Mututal Information (NMI) is a common metric

→ Normalized between 0 (no mutual information) to 1 (perfect correlation)

$$\mathrm{NMI}(U, V) = \frac{\mathrm{MI}(U, V)}{\mathrm{mean}(\mathrm{H}(U) + \mathrm{H}(V))}$$

Evaluation Metric

→ We evaluate constructed clusters by comparing to Ground Truth labels

→ Normalized Mututal Information (NMI) is a common metric

→ Normalized between 0 (no mutual information) to 1 (perfect correlation)

Mutual Information

$$\mathrm{NMI}(U, V) = \frac{\mathrm{MI}(U, V)}{\mathrm{mean}(\mathrm{H}(U) + \mathrm{H}(V))}$$

Evaluation Metric

→ We evaluate constructed clusters by comparing to Ground Truth labels

→ Normalized Mututal Information (NMI) is a common metric

→ Normalized between 0 (no mutual information) to 1 (perfect correlation)

Mutual Information

$$\mathrm{NMI}(U, V) = \frac{\mathrm{MI}(U, V)}{\mathrm{mean}(\mathrm{H}(U) + \mathrm{H}(V))}$$

Entropy of Clusters

Evaluation Metric

→ We evaluate constructed clusters by comparing to Ground Truth labels

→ Normalized Mututal Information (NMI) is a common metric

→ Normalized between 0 (no mutual information) to 1 (perfect correlation)

→ However, NMI is not corrected for chance. Simply increasing number of clusters can increase NMI

Evaluation Metric

→ We evaluate constructed clusters by comparing to Ground Truth labels

→ Normalized Mututal Information (NMI) is a common metric

→ Normalized between 0 (no mutual information) to 1 (perfect correlation)

→ However, NMI is not corrected for chance. Simply increasing number of clusters can increase NMI

→ Use Adjusted Mutual Information (AMI) instead

$$\mathrm{AMI}(U, V) = \frac{\mathrm{MI}(U, V) - \mathbb{E}[\mathrm{MI}(U, V)]}{\mathrm{mean}(\mathrm{H}(U) + \mathrm{H}(V)) - \mathbb{E}[\mathrm{MI}(U, V)]}$$

Hyperparameter Search

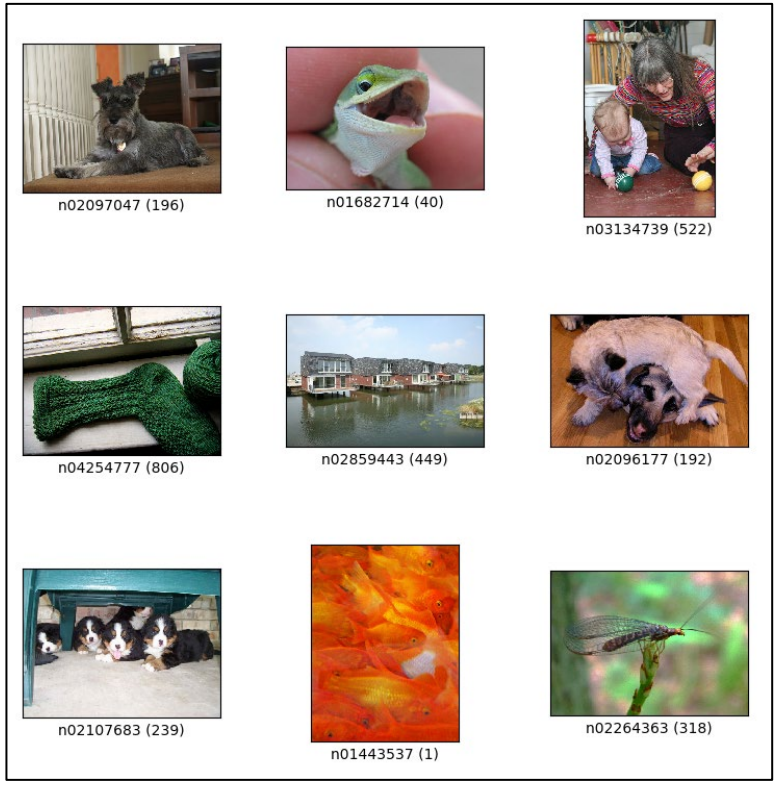
Hyperparameter Search

→ HPs selectected over three ImageNet-based datasets

Hyperparameter Search

→ HPs selectected over three ImageNet-based datasets

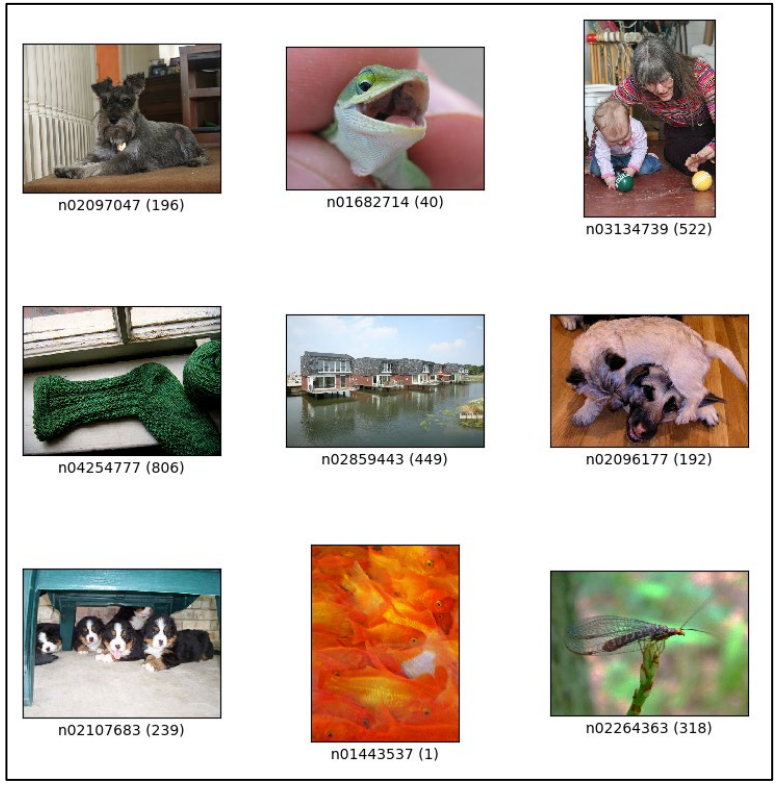
ImageNet-1K



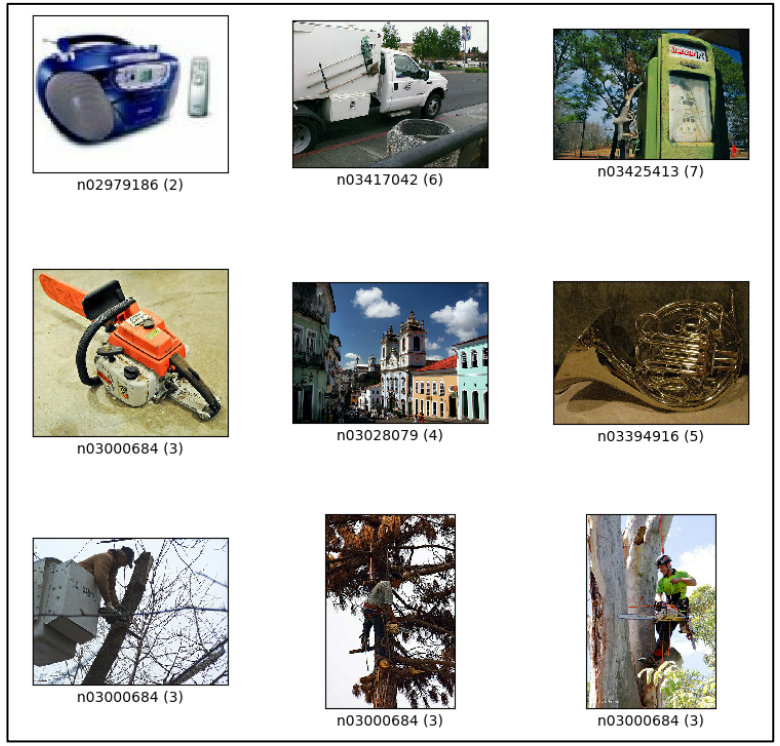
Hyperparameter Search

→ HPs selectected over three ImageNet-based datasets

ImageNet-1K



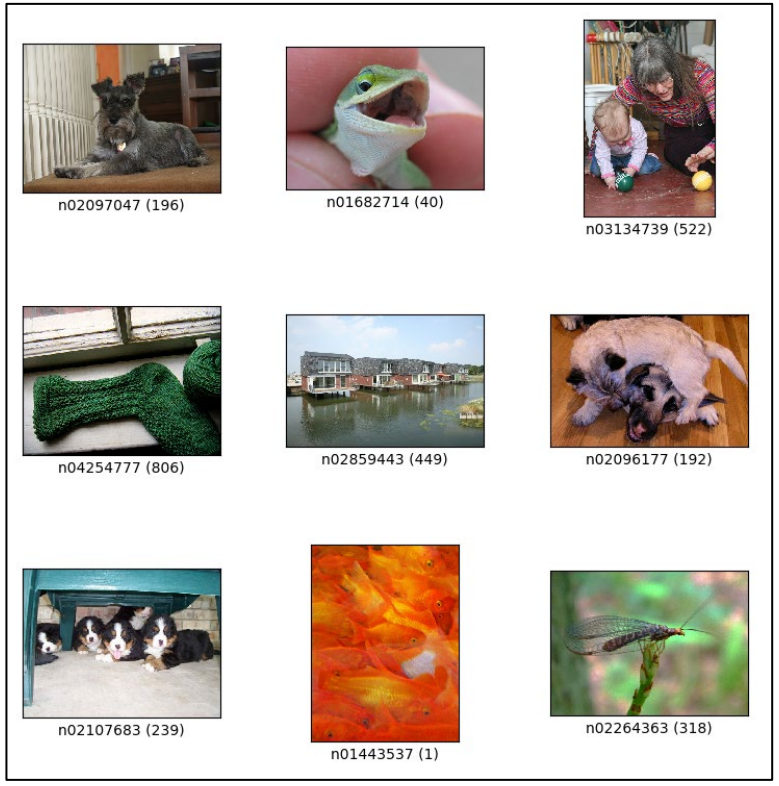
Imagenette



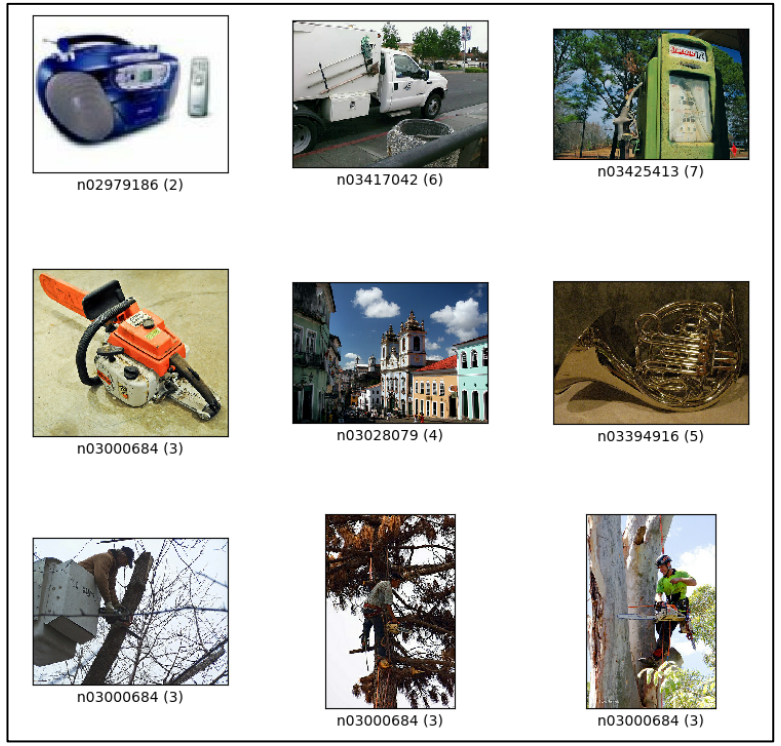
Hyperparameter Search

→ HPs selectected over three ImageNet-based datasets

ImageNet-1K



Imagenette



Imagewoof

Hyperparameter Search

→ HPs selectected over three ImageNet-based datasets

→ Create validation splits via class-stratified sampling

Hyperparameter Search

→ HPs selectected over three ImageNet-based datasets

→ Create validation splits via class-stratified sampling

→ First search dimensionality reduction
→ Then line-search clustering method hyperparameters

Hyperparameter Search

→ HPs selectected over three ImageNet-based datasets

→ Create validation splits via class-stratified sampling

→ First search dimensionality reduction
→ Then line-search clustering method hyperparameters

→ All clustering methods (except Spectral) performs best with UMAP
→ Reducing to 5-200 dimensions leads to equal performance

Datasets

Datasets

Type	Dataset	Reference	# Sample	# Class	ρ	Description
In-Domain	ImageNet-1k	Russakovsky et al. (2015)	50 000	1 000	1.00	Diverse general objects
	ImageNet-v2	Recht et al. (2019)	10 000	1 000	1.00	Diverse general objects
	CIFAR-10	Krizhevsky (2009)	10 000	10	1.00	Diverse general objects
	CIFAR-100	Krizhevsky (2009)	10 000	100	1.00	Diverse general objects
	ImageNet-9 originals	Xiao et al. (2020)	4 050	9	1.00	Diverse general objects

Datasets

Type	Dataset	Reference	# Sample	# Class	ρ	Description
In-Domain	ImageNet-1k	Russakovsky et al. (2015)	50 000	1 000	1.00	Diverse general objects
	ImageNet-v2	Recht et al. (2019)	10 000	1 000	1.00	Diverse general objects
	CIFAR-10	Krizhevsky (2009)	10 000	10	1.00	Diverse general objects
	CIFAR-100	Krizhevsky (2009)	10 000	100	1.00	Diverse general objects
	ImageNet-9 originals	Xiao et al. (2020)	4 050	9	1.00	Diverse general objects
Domain-shift	ImageNet-9 FG-only	Xiao et al. (2020)	4 050	9	1.00	Isolated foregrounds
	ImageNet-9 MixRand	Xiao et al. (2020)	4 050	9	1.00	Remixed fore/background
	ImageNet-R	Hendrycks et al. (2021a)	30 000	200	8.43	Art/sculptures of objects
	ImageNet-Sketch	Wang et al. (2019)	50 889	1 000	1.02	Sketches of objects

Datasets

Type	Dataset	Reference	# Sample	# Class	ρ	Description
In-Domain	ImageNet-1k	Russakovsky et al. (2015)	50 000	1 000	1.00	Diverse general objects
	ImageNet-v2	Recht et al. (2019)	10 000	1 000	1.00	Diverse general objects
	CIFAR-10	Krizhevsky (2009)	10 000	10	1.00	Diverse general objects
	CIFAR-100	Krizhevsky (2009)	10 000	100	1.00	Diverse general objects
	ImageNet-9 originals	Xiao et al. (2020)	4 050	9	1.00	Diverse general objects
Domain-shift	ImageNet-9 FG-only	Xiao et al. (2020)	4 050	9	1.00	Isolated foregrounds
	ImageNet-9 MixRand	Xiao et al. (2020)	4 050	9	1.00	Remixed fore/background
	ImageNet-R	Hendrycks et al. (2021a)	30 000	200	8.43	Art/sculptures of objects
	ImageNet-Sketch	Wang et al. (2019)	50 889	1 000	1.02	Sketches of objects
Near-OOD	ImageNet-O	Hendrycks et al. (2021b)	2 000	200	6.00	Diverse general objects
	LSUN	Yu et al. (2015)	10 000	10	1.00	Urban/indoor scenes
	Places365	Zhou et al. (2018)	36 500	365	1.00	Scenes

Datasets

Type	Dataset	Reference	# Sample	# Class	ρ	Description
In-Domain	ImageNet-1k	Russakovsky et al. (2015)	50 000	1 000	1.00	Diverse general objects
	ImageNet-v2	Recht et al. (2019)	10 000	1 000	1.00	Diverse general objects
	CIFAR-10	Krizhevsky (2009)	10 000	10	1.00	Diverse general objects
	CIFAR-100	Krizhevsky (2009)	10 000	100	1.00	Diverse general objects
	ImageNet-9 originals	Xiao et al. (2020)	4 050	9	1.00	Diverse general objects
Domain-shift	ImageNet-9 FG-only	Xiao et al. (2020)	4 050	9	1.00	Isolated foregrounds
	ImageNet-9 MixRand	Xiao et al. (2020)	4 050	9	1.00	Remixed fore/background
	ImageNet-R	Hendrycks et al. (2021a)	30 000	200	8.43	Art/sculptures of objects
	ImageNet-Sketch	Wang et al. (2019)	50 889	1 000	1.02	Sketches of objects
Near-OOD	ImageNet-O	Hendrycks et al. (2021b)	2 000	200	6.00	Diverse general objects
	LSUN	Yu et al. (2015)	10 000	10	1.00	Urban/indoor scenes
	Places365	Zhou et al. (2018)	36 500	365	1.00	Scenes
Fine-grained	FGVC Aircraft	Maji et al. (2013)	3 333	100	1.03	Aircraft variants
	Stanford Cars	Krause et al. (2013)	8 041	196	2.83	Car variants
	Oxford Flowers	Nilsback and Zisserman (2008)	6 149	102	11.90	Flower variants
	NABirds	Van Horn et al. (2015)	24 633	555	6.67	Bird species
	BIOSCAN-1M	Gharaee et al. (2023)	24 799	2 688	782.50	Insect species
	iNaturalist-2021	Van Horn et al. (2021)	100 000	10 000	1.00	Plant & animal species

Datasets

Type	Dataset	Reference	# Sample	# Class	ρ	Description
In-Domain	ImageNet-1k	Russakovsky et al. (2015)	50 000	1 000	1.00	Diverse general objects
	ImageNet-v2	Recht et al. (2019)	10 000	1 000	1.00	Diverse general objects
	CIFAR-10	Krizhevsky (2009)	10 000	10	1.00	Diverse general objects
	CIFAR-100	Krizhevsky (2009)	10 000	100	1.00	Diverse general objects
	ImageNet-9 originals	Xiao et al. (2020)	4 050	9	1.00	Diverse general objects
Domain-shift	ImageNet-9 FG-only	Xiao et al. (2020)	4 050	9	1.00	Isolated foregrounds
	ImageNet-9 MixRand	Xiao et al. (2020)	4 050	9	1.00	Remixed fore/background
	ImageNet-R	Hendrycks et al. (2021a)	30 000	200	8.43	Art/sculptures of objects
	ImageNet-Sketch	Wang et al. (2019)	50 889	1 000	1.02	Sketches of objects
Near-OOD	ImageNet-O	Hendrycks et al. (2021b)	2 000	200	6.00	Diverse general objects
	LSUN	Yu et al. (2015)	10 000	10	1.00	Urban/indoor scenes
	Places365	Zhou et al. (2018)	36 500	365	1.00	Scenes
Fine-grained	FGVC Aircraft	Maji et al. (2013)	3 333	100	1.03	Aircraft variants
	Stanford Cars	Krause et al. (2013)	8 041	196	2.83	Car variants
	Oxford Flowers	Nilsback and Zisserman (2008)	6 149	102	11.90	Flower variants
	NABirds	Van Horn et al. (2015)	24 633	555	6.67	Bird species
	BIOSCAN-1M	Gharaee et al. (2023)	24 799	2 688	782.50	Insect species
	iNaturalist-2021	Van Horn et al. (2021)	100 000	10 000	1.00	Plant & animal species
Far-OOD	CelebA	Liu et al. (2015)	19 962	1 000	32.00	Human faces (identity)
	UTKFace	Zhang et al. (2017)	5 925	101	549.00	Human faces (age)
	BreakHis	Spanhol et al. (2016)	3 164	32	8.60	Tumor tissue microscopy
	DTD	Cimpoi et al. (2014)	1 880	47	1.00	Texture descriptions
	EuroSAT	Helber et al. (2019)	4 050	10	1.50	Satellite RGB images
	MNIST	LeCun et al. (1998)	10 000	10	1.27	Handwritten digits
	Fashion MNIST	Xiao et al. (2017)	10 000	10	1.00	Clothing articles
	SVHN	Netzer et al. (2011)	26 032	10	3.20	House numbers

Results

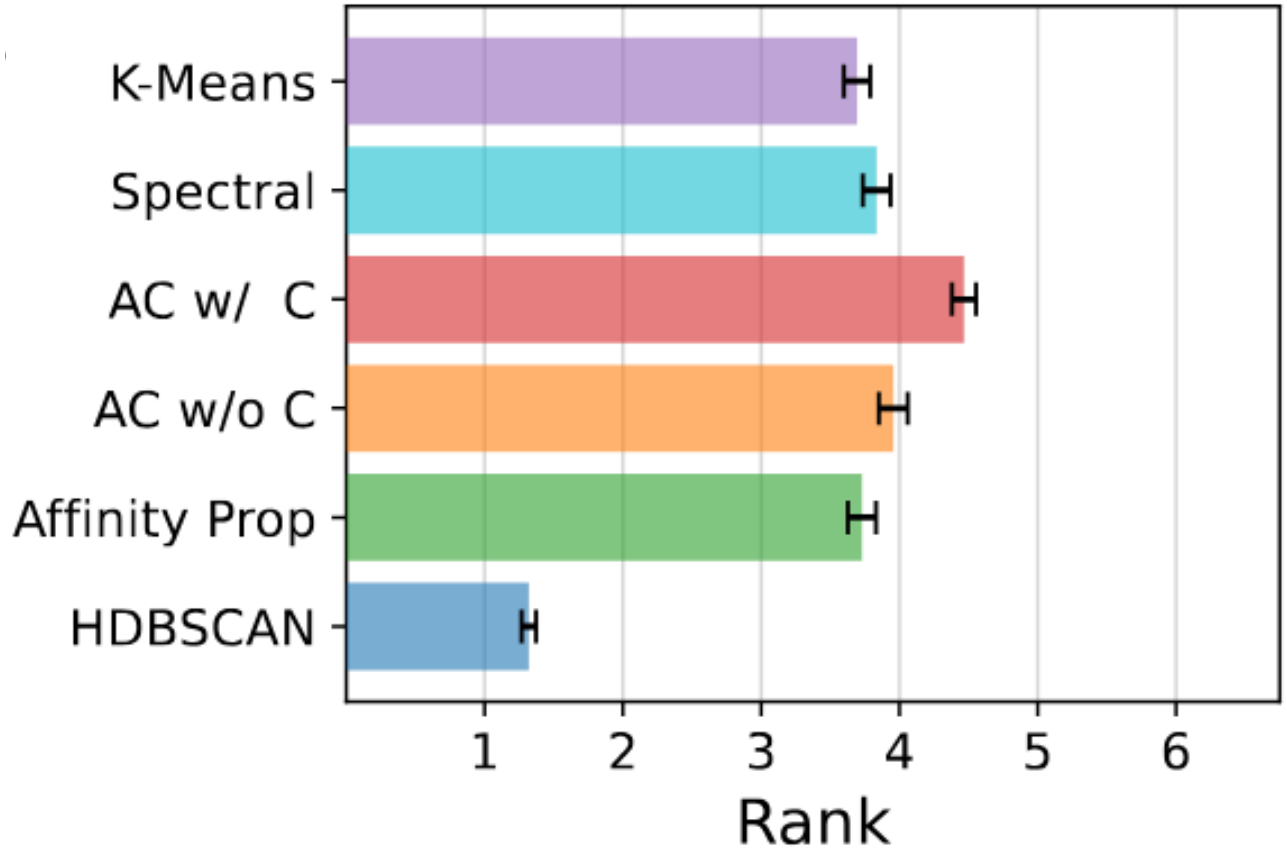
Which clustering method is best?

Which clustering method is best?

→ Rank each clustering method per dataset for each encoder
→ Rank 1 = worst, Rank 6 = best

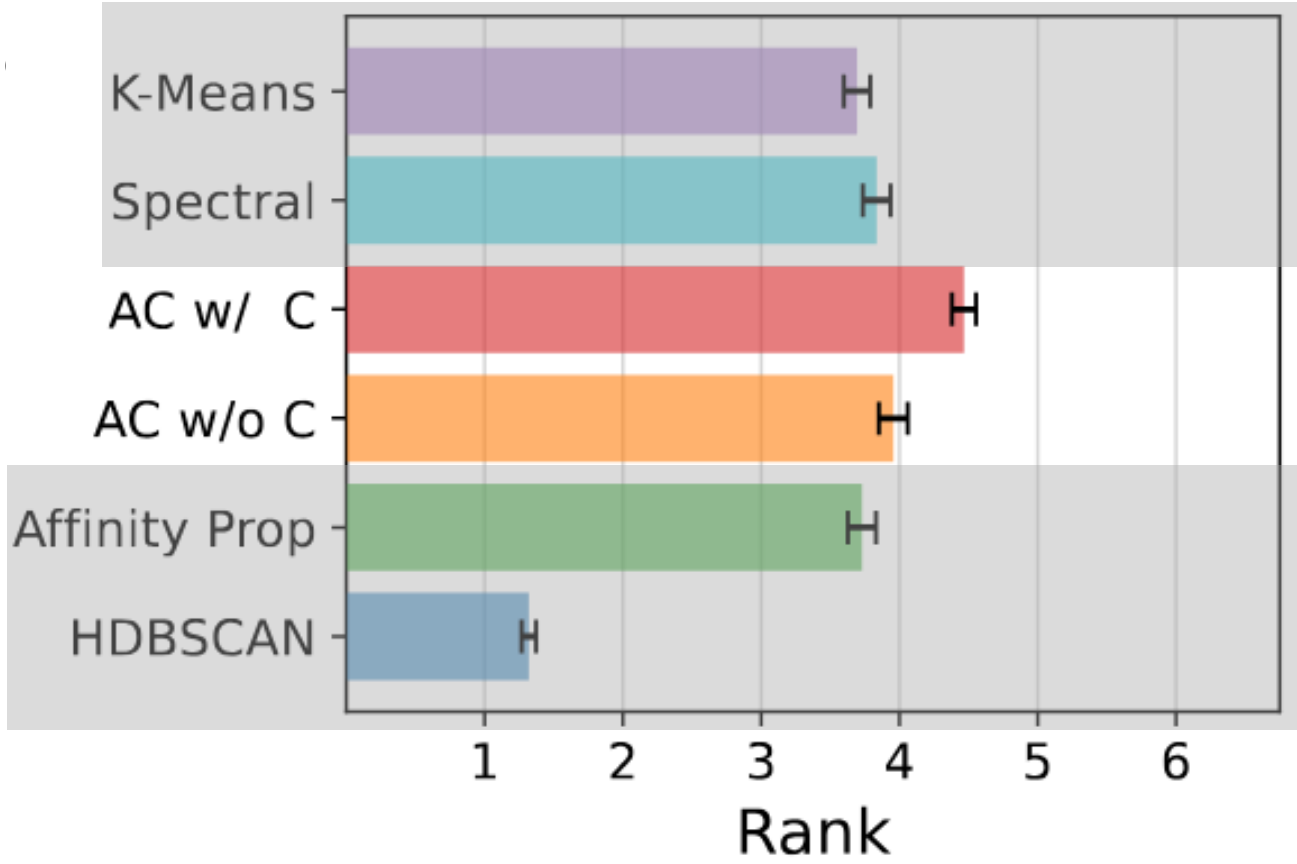
Which clustering method is best?

→ Rank each clustering method per dataset for each
→ Rank 1 = worst, Rank 6 = best



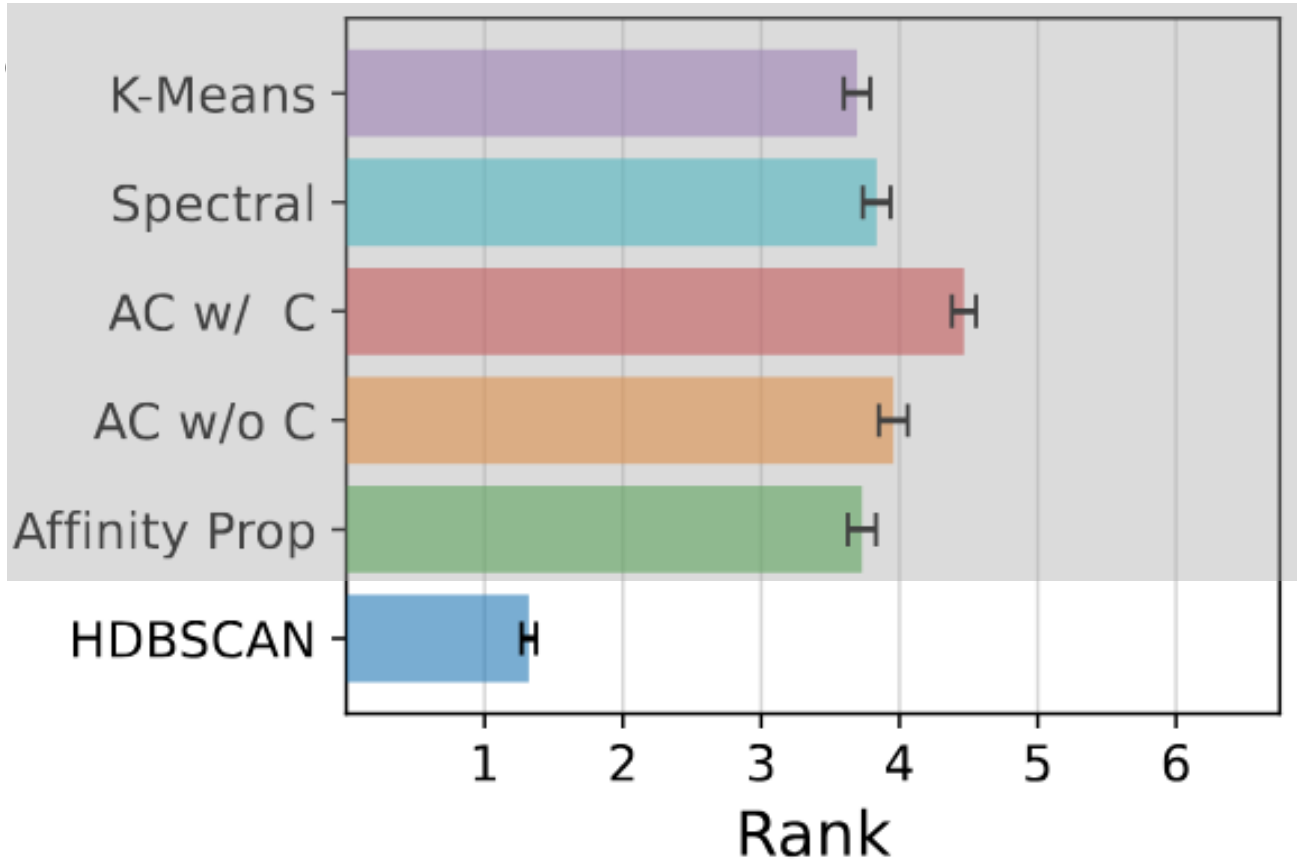
Which clustering method is best?

→ Rank each clustering method per dataset for each
→ Rank 1 = worst, Rank 6 = best



Which clustering method is best?

→ Rank each clustering method per dataset for each
→ Rank 1 = worst, Rank 6 = best



Comparing Supervised & SSL Pretrained Encoders

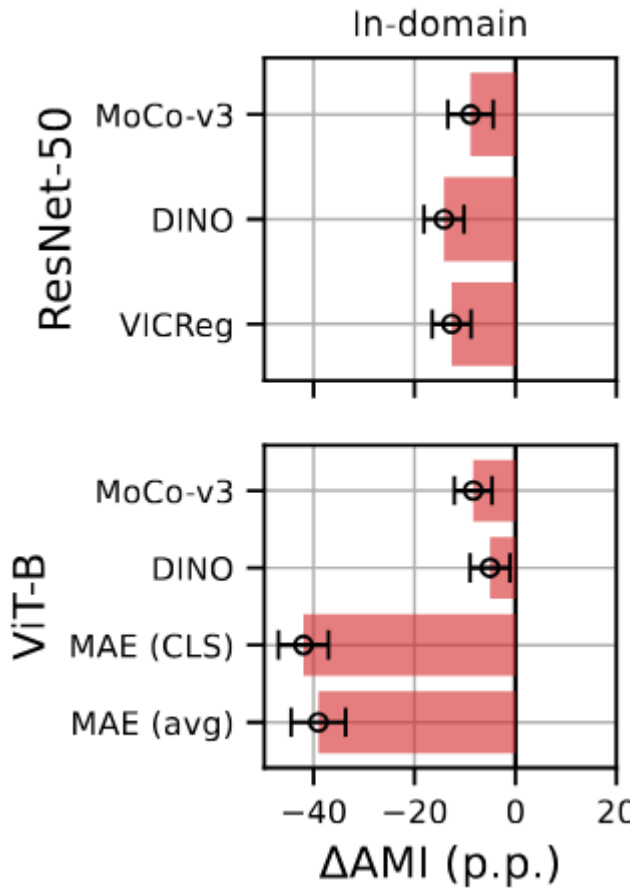
Comparing Supervised & SSL Pretrained Encoders

→ We compare the effect of SSL pretraining to fully-supervised Cross Entropy models

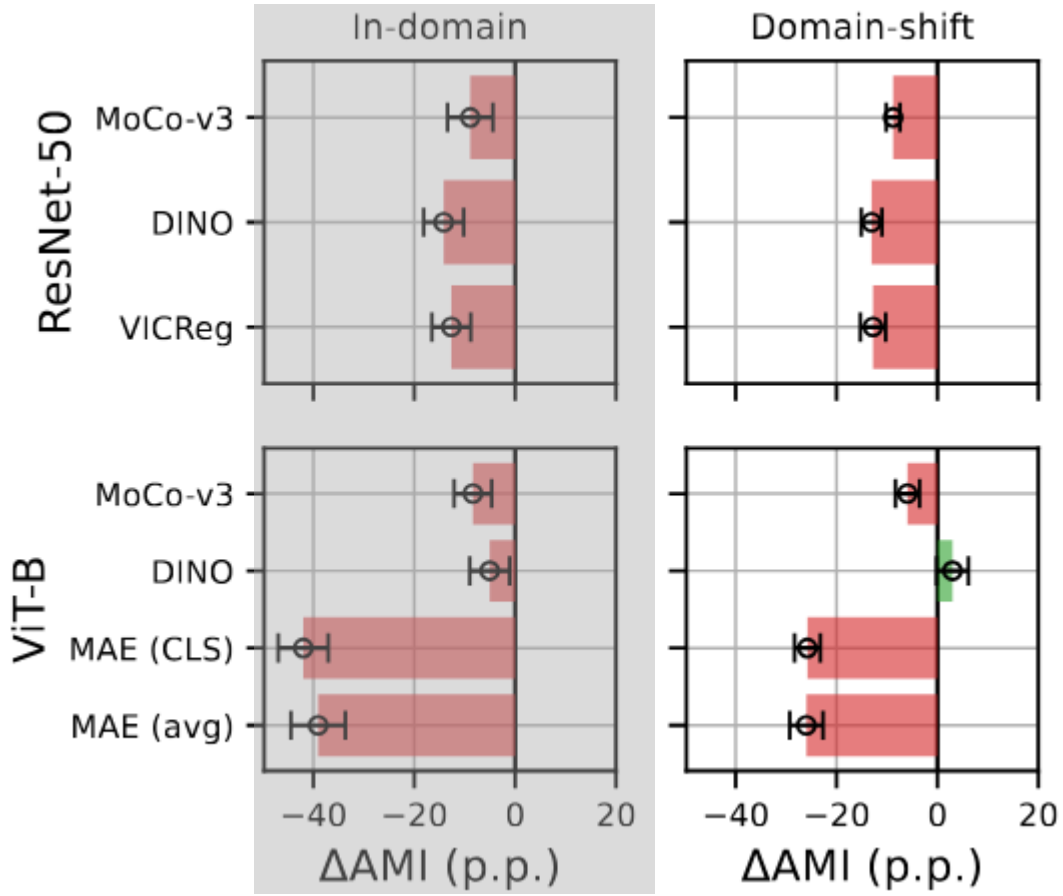
→ Measuered as difference in AMI scores (Δ AMI)

SSL Pretrained Encoders

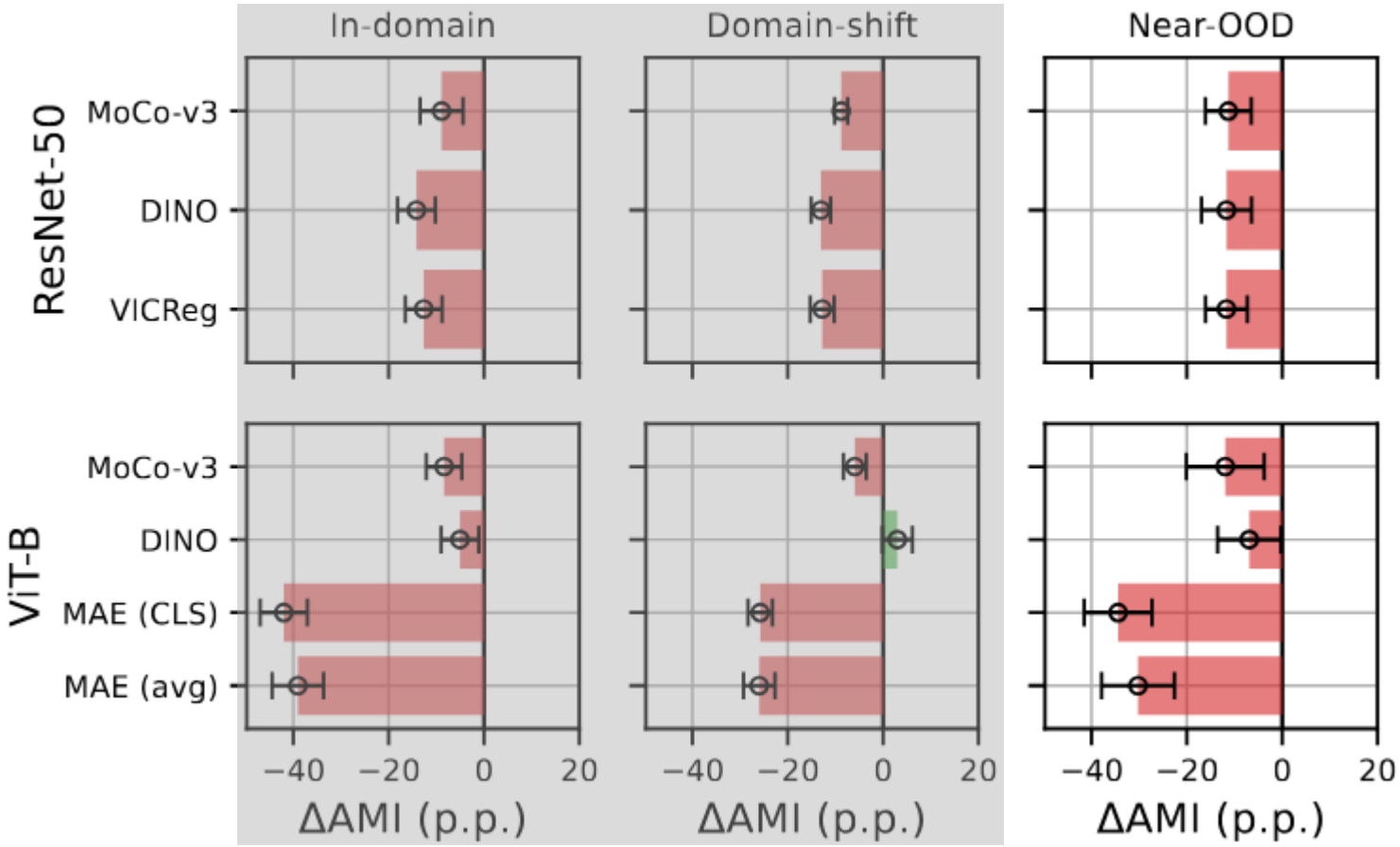
SSL Pretrained Encoders



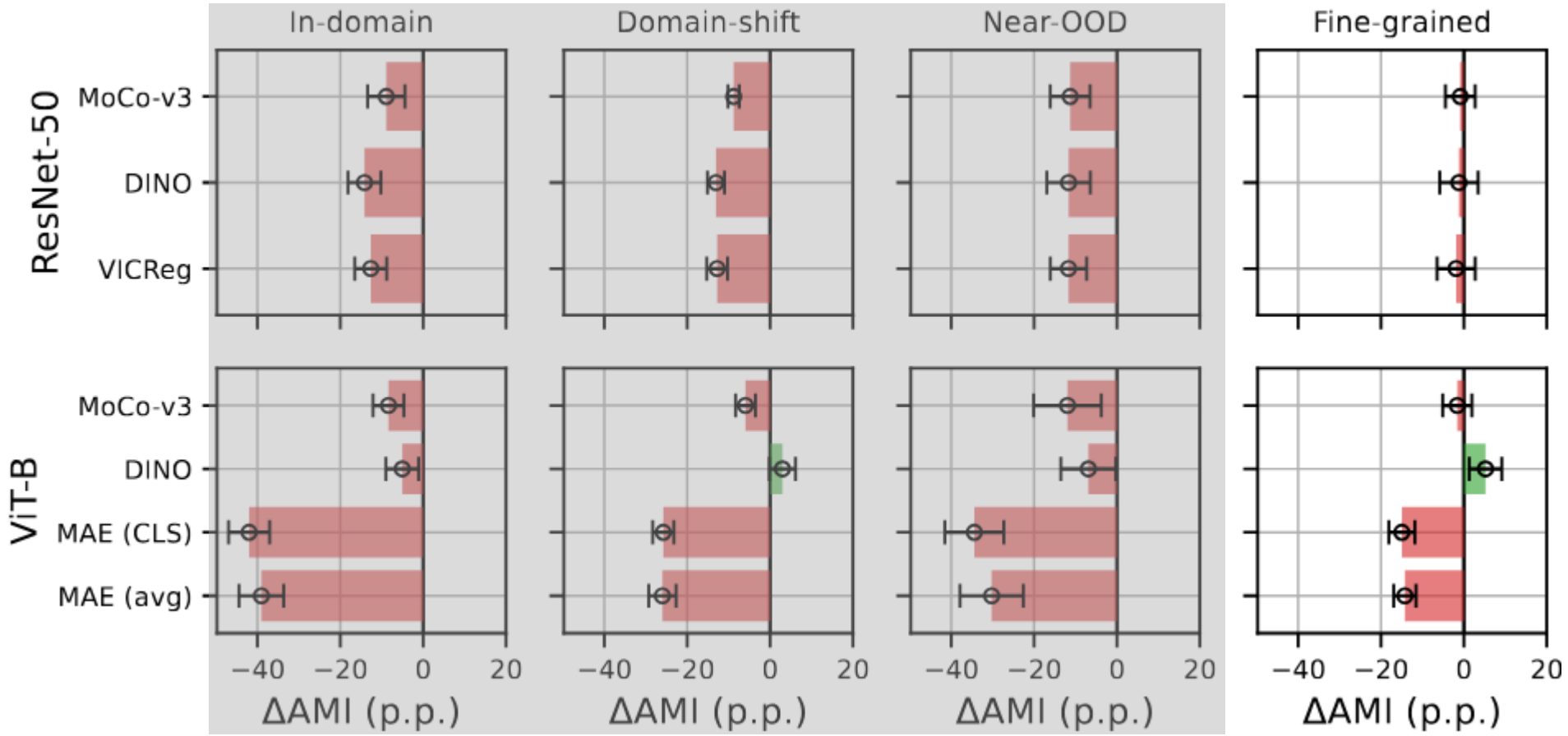
SSL Pretrained Encoders



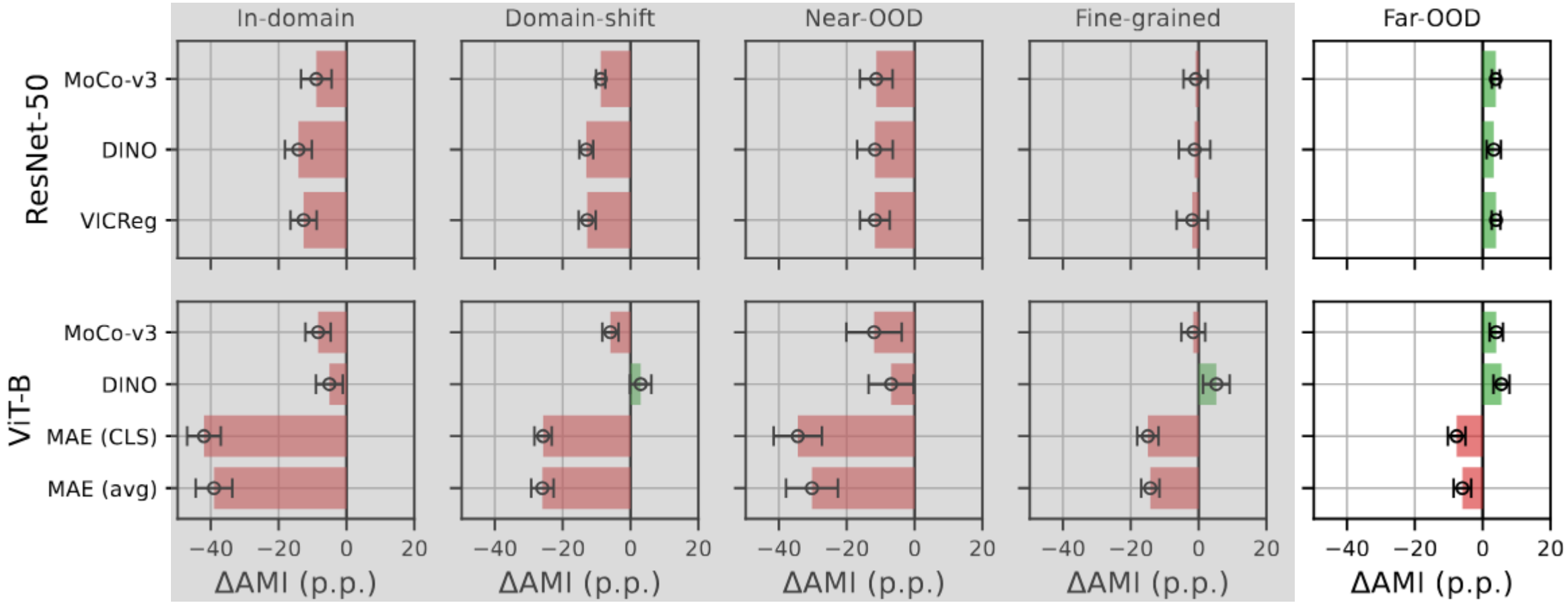
SSL Pretrained Encoders



SSL Pretrained Encoders

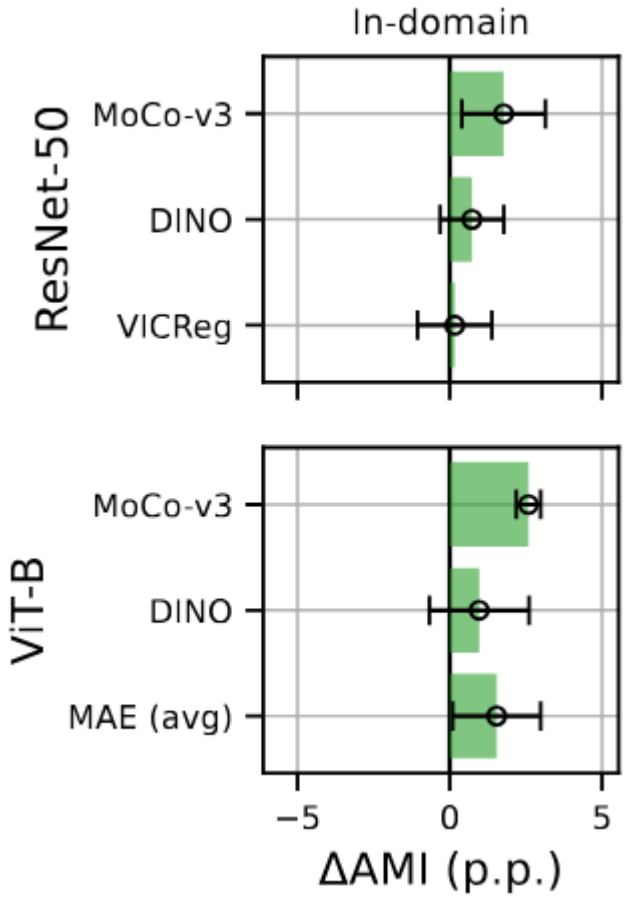


SSL Pretrained Encoders

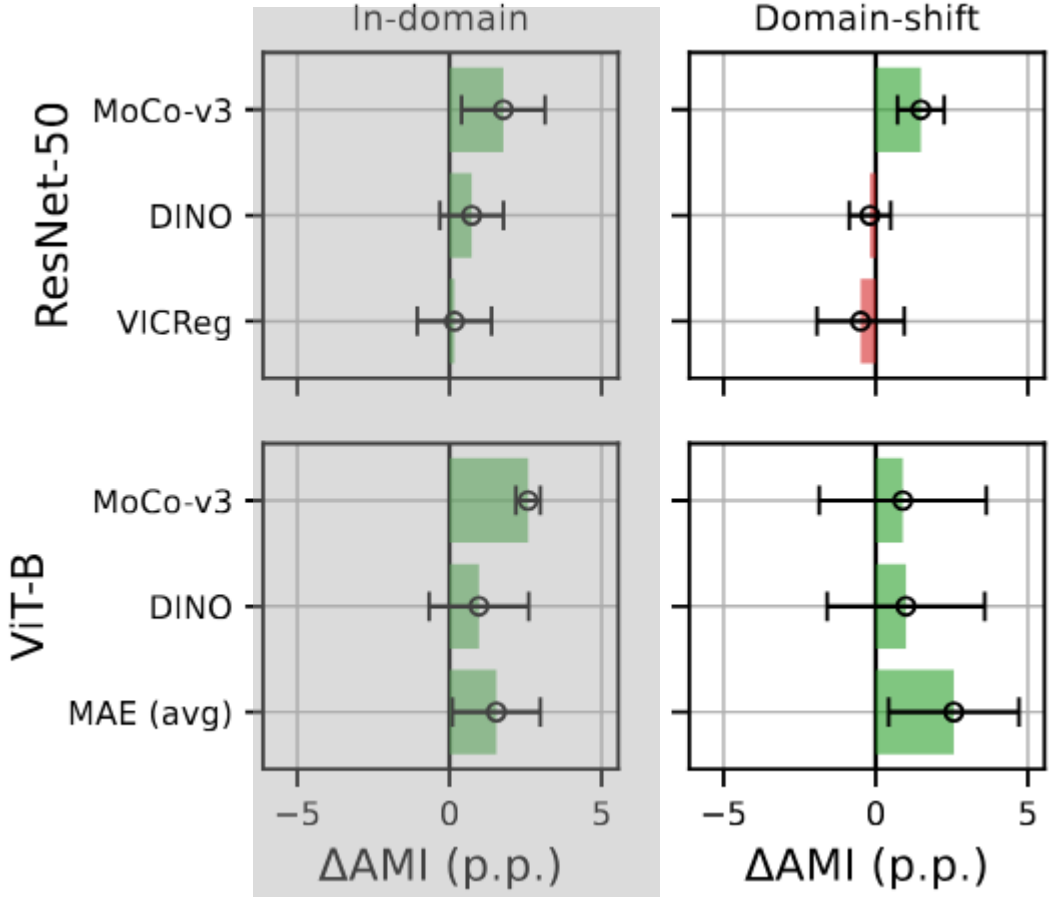


SSL Fine-tuned Encoders

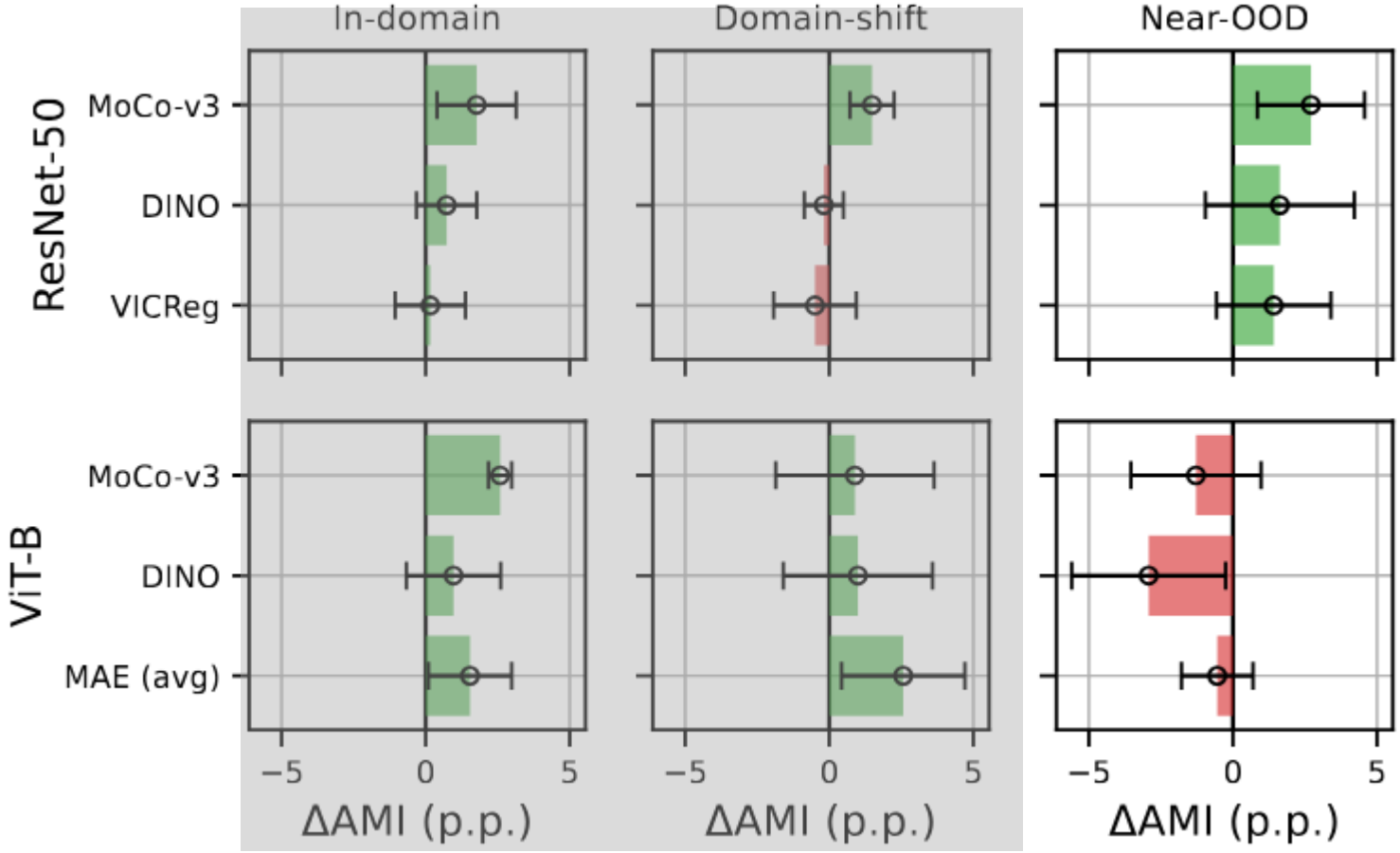
SSL Fine-tuned Encoders



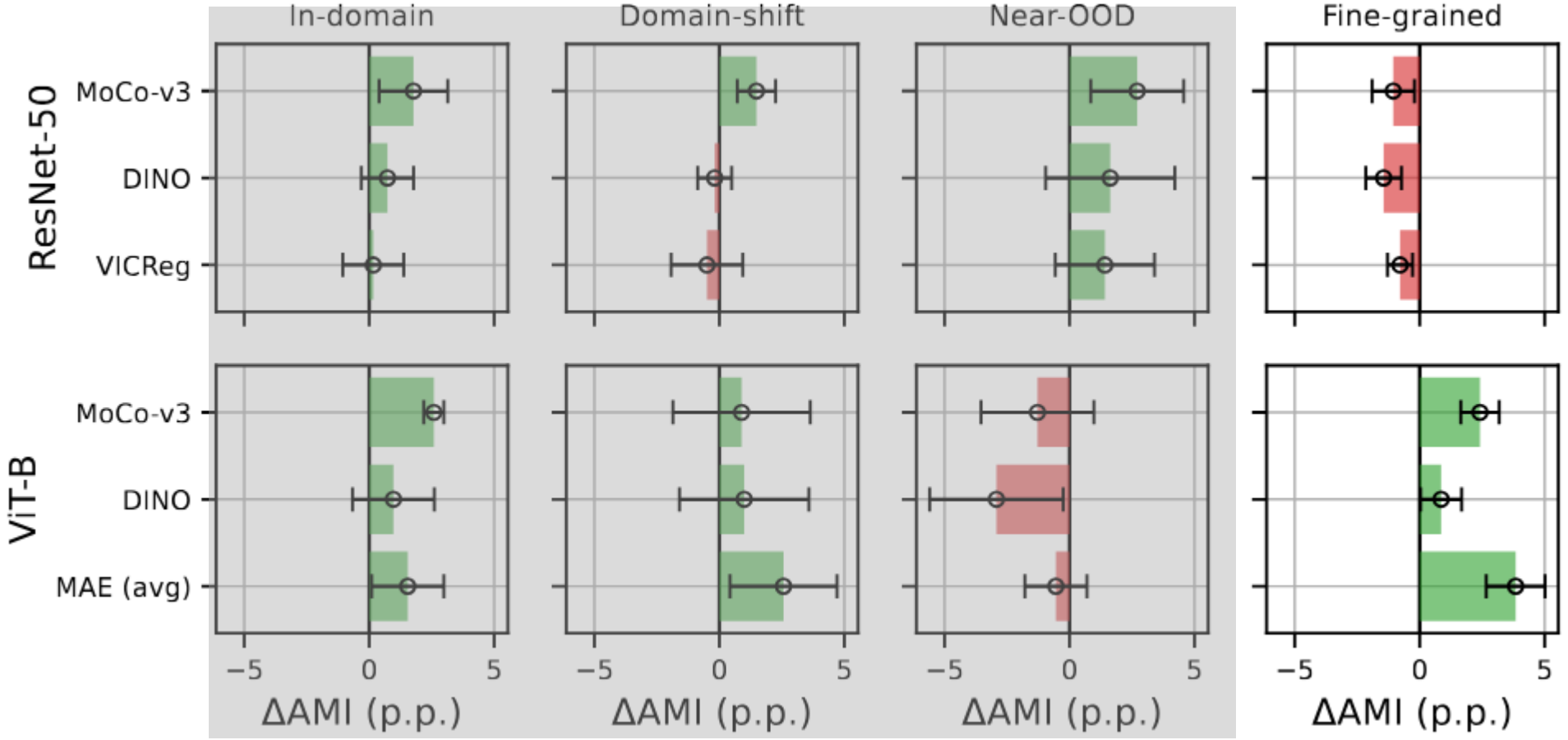
SSL Fine-tuned Encoders



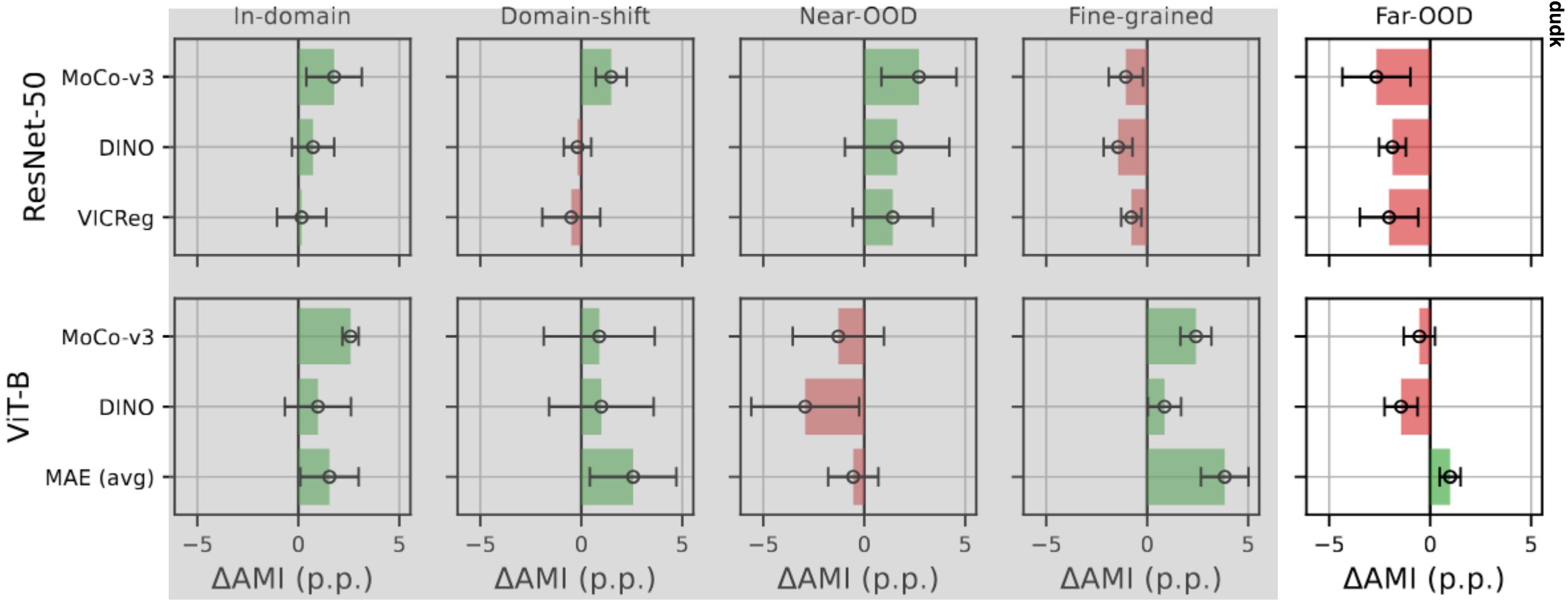
SSL Fine-tuned Encoders



SSL Fine-tuned Encoders



SSL Fine-tuned Encoders



Taxonomic Hierarchies

Taxonomic Hierarchies

→ How well does the clusters match the taxonomic hierarchies?

Taxonomic Hierarchies

→ How well does the clusters match the taxonomic hierarchies?

$$\mathrm{AMI}(\mathrm{Y}_{\mathrm{True}}, \mathrm{Y}_{\mathrm{Pred}}) = \frac{\mathrm{MI}(\mathrm{Y}_{\mathrm{True}}, \mathrm{Y}_{\mathrm{Pred}}) - \mathbb{E}[\mathrm{MI}(\mathrm{Y}_{\mathrm{True}}, \mathrm{Y}_{\mathrm{Pred}})]}{\mathrm{mean}\big(\mathrm{H}(Y_{\mathrm{True}}) + \mathrm{H}(Y_{\mathrm{Pred}})\big) - \mathbb{E}[\mathrm{MI}(\mathrm{Y}_{\mathrm{True}}, \mathrm{Y}_{\mathrm{Pred}})}$$

Taxonomic Hierarchies

→ How well does the clusters match the taxonomic hierarchies?

$$\mathrm{AMI}(\mathrm{Y_{True}}, \mathrm{Y_{Pred}}) = \frac{\mathrm{MI}(\mathrm{Y_{True}}, \mathrm{Y_{Pred}}) - \mathbb{E}[\mathrm{MI}(\mathrm{Y_{True}}, \mathrm{Y_{Pred}})]}{\mathrm{H}(\mathrm{Y_{True}}) - \mathbb{E}[\mathrm{MI}(\mathrm{Y_{True}}, \mathrm{Y_{Pred}})}$$

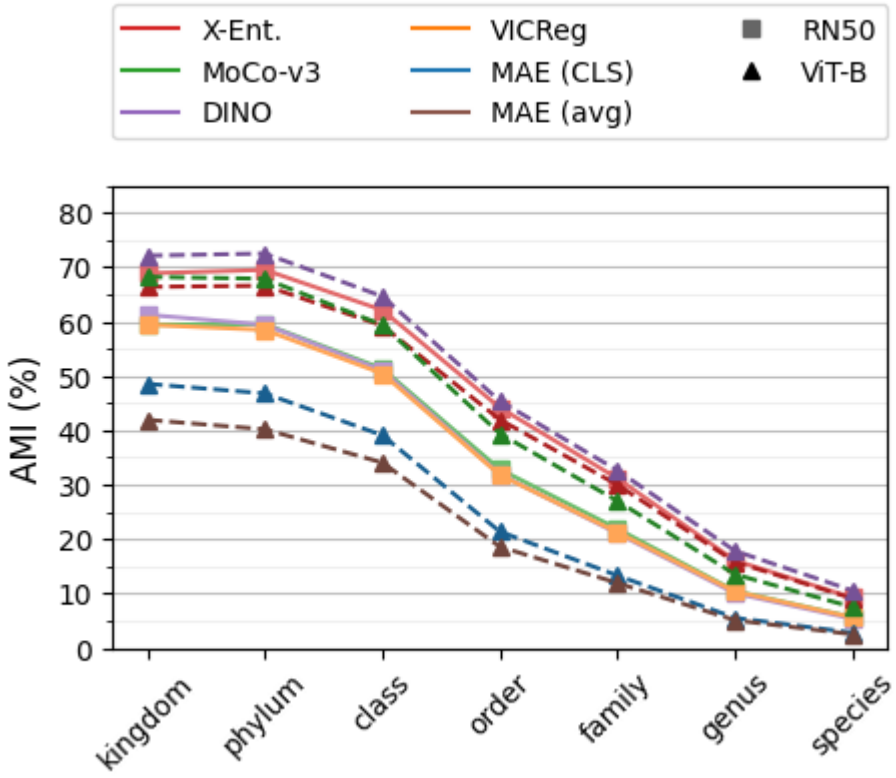
Taxonomic Hierarchies

→ How well does the clusters match the taxonomic hierarchies?

→ Now describes the percentage of entropy in the label that is explained by observing the prediction

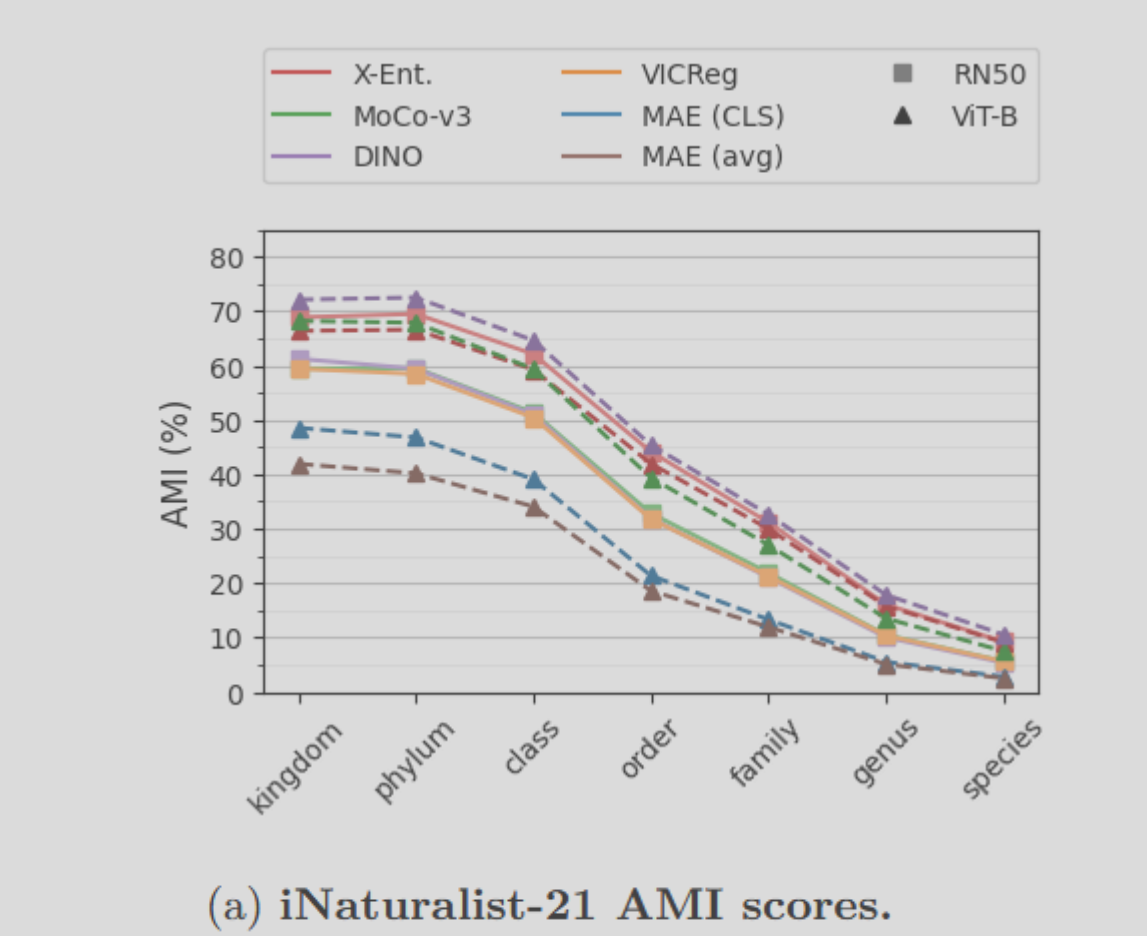
$$\mathrm{AMI}(\mathrm{Y_{True}}, \mathrm{Y_{Pred}}) = \frac{\mathrm{MI}(\mathrm{Y_{True}}, \mathrm{Y_{Pred}}) - \mathbb{E}[\mathrm{MI}(\mathrm{Y_{True}}, \mathrm{Y_{Pred}})]}{\mathrm{H}(\mathrm{Y_{True}}) - \mathbb{E}[\mathrm{MI}(\mathrm{Y_{True}}, \mathrm{Y_{Pred}})]}$$

Taxonomic Hierarchies

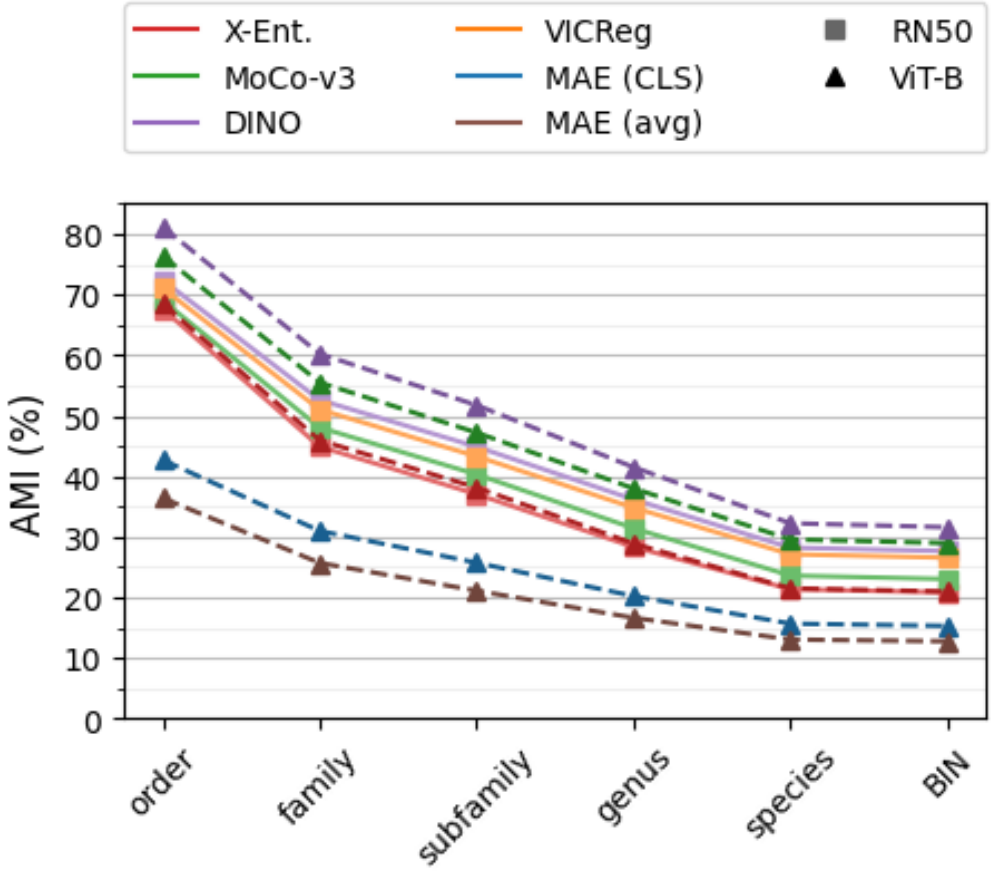


(a) **iNaturalist-21 AMI scores.**

Taxonomic Hierarchies



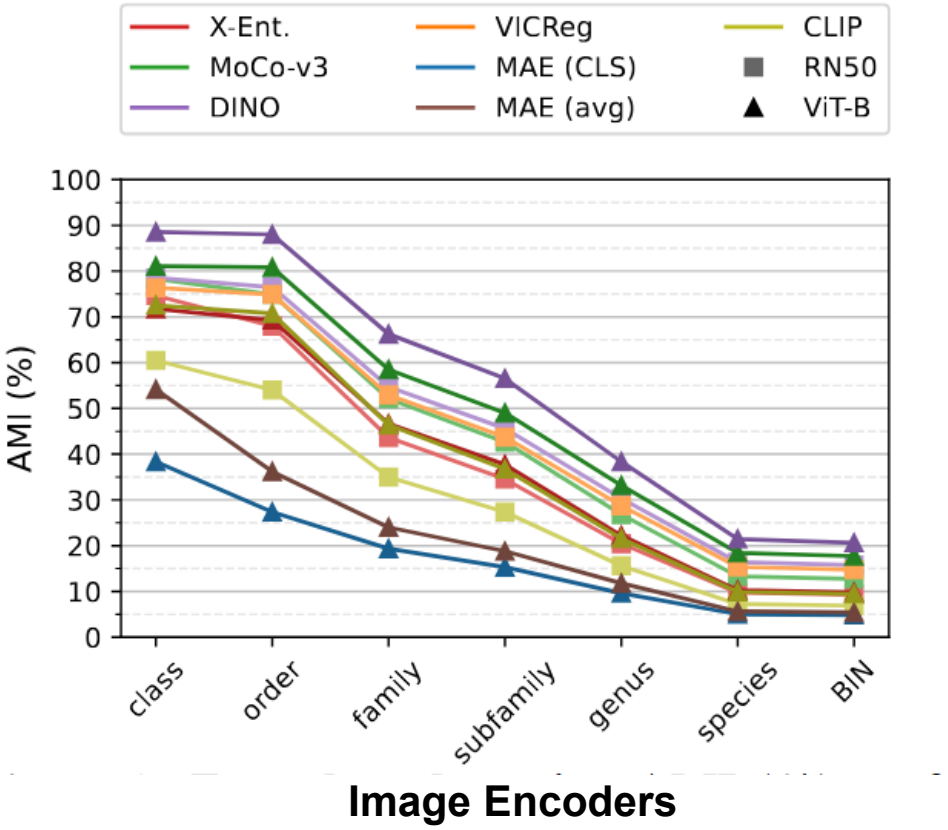
(a) iNaturalist-21 AMI scores.



(b) BIOSCAN-1M AMI scores.

Taxonomic Hierarchies – BIOSCAN-5M

Taxonomic Hierarchies – BIOSCAN-5M



Taxonomic Hierarchies – BIOSCAN-5M

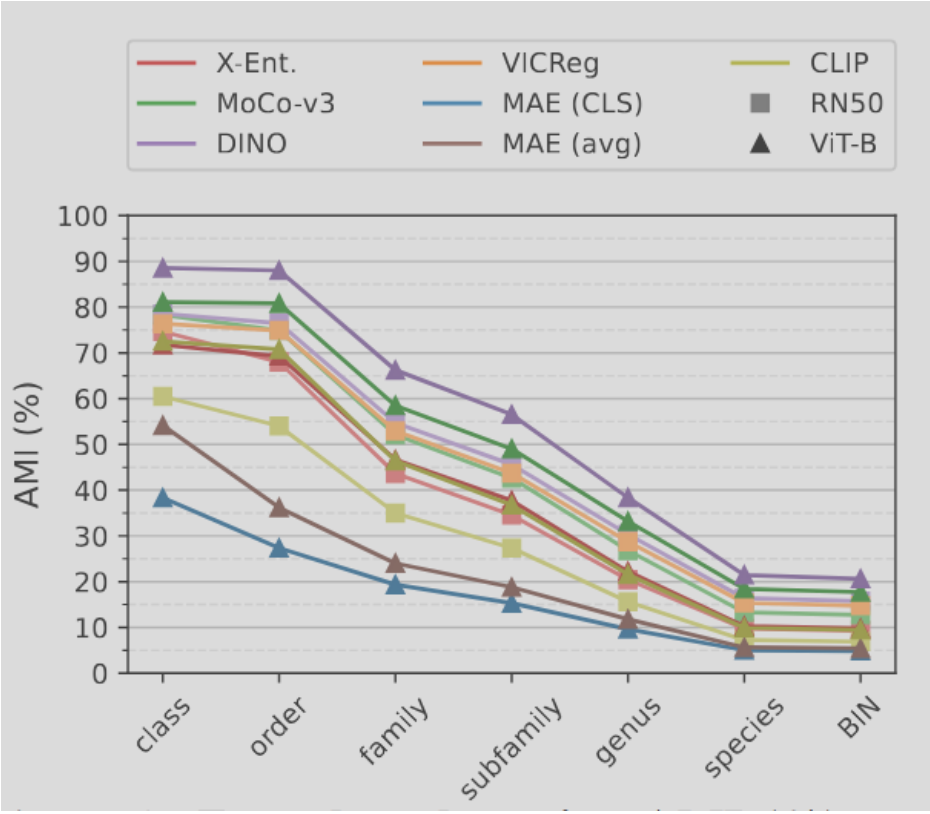
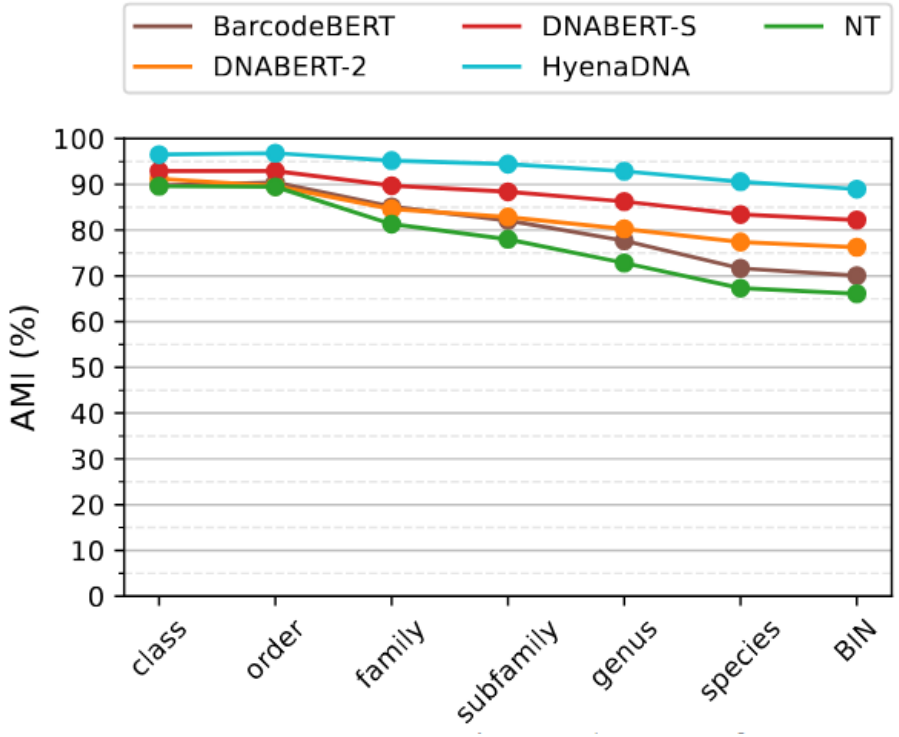


Image Encoders



DNA Encoders

Potential of ZSC Evaluations

How does it compare to probing?

How does it compare to probing?

→ Probing is a way of converting a SSL pretained encoder into a classifiers

→ Typically done via kNN prbing or linear probing

How does it compare to probing?

→ Probing is a way of converting a SSL pretained encoder into a classifiers

→ Typically done via kNN prbing or linear probing

→ kNN probing = for each test point assign label based on k closest samples from a labeled dataset

→ Linear probing = train a Linear layer on top of the frozen backbone (ie logistic regression)

How does it compare to probing?

→ Probing is a way of converting a SSL pretained encoder into a classifiers

→ Typically done via kNN prbing or linear probing

→ kNN probing = for each test point assign label based on k closest samples from a labeled dataset

→ Linear probing = train a Linear layer on top of the frozen backbone (ie logistic regression)

→ We compare Spearman's Correlation between AMI and kNN accuracy

How does it compare to probing?

Arch.	Clusterer	$k=1$	$k=10$	$k=20$	$k=100$	$k=200$
RN50	K-Means	51	48	48	48	49
	Spectral	54	52	52	53	53
	AC w/ C	49	47	47	47	47
	AC w/o C	45	44	44	44	44
	Affinity Prop.	49	47	47	47	47
	HDBSCAN	51	49	49	49	49
ViT-B	K-Means	37	37	37	39	39
	Spectral	37	38	38	40	40
	AC w/ C	33	34	34	35	36
	AC w/o C	39	40	40	41	41
	Affinity Prop.	36	37	36	38	38
	HDBSCAN	38	39	38	40	40

* Correlation scores multiplied by 100 for readability

How does it compare to probing?

→ AMI and kNN accuracy is only moderately correlated

→ ZSC can therefore be seen as an orthogonal evaluation process to probing

Arch.	Clusterer	$k=1$	$k=10$	$k=20$	$k=100$	$k=200$
RN50	K-Means	51	48	48	48	49
	Spectral	54	52	52	53	53
	AC w/ C	49	47	47	47	47
	AC w/o C	45	44	44	44	44
	Affinity Prop.	49	47	47	47	47
	HDBSCAN	51	49	49	49	49
ViT-B	K-Means	37	37	37	39	39
	Spectral	37	38	38	40	40
	AC w/ C	33	34	34	35	36
	AC w/o C	39	40	40	41	41
	Affinity Prop.	36	37	36	38	38
	HDBSCAN	38	39	38	40	40

* Correlation scores multiplied by 100 for readability

What if you don’t have Ground Truth?

What if you don’t have Ground Truth?

→ But what if you don’t have a labeled dataset?

What if you don’t have Ground Truth?

→ But what if you don’t have a labeled dataset?

→ Cluster quality can be measured using the *Silhouette score*

→ The Silhouette score is an *intrinsic* measure based on how ”tight” and separated clusters are

What if you don’t have Ground Truth?

→ But what if you don’t have a labeled dataset?

→ Cluster quality can be measured using the *Silhouette score*

→ The Silhouette score is an *intrinsic* measure based on how ”tight” and separated clusters are

$$S = \frac{1}{N} \sum_{i}^{N} \frac{a_i - b_i}{\max(a_i, b_i)}$$

a_i $-$ *Mean within cluster distance*
b_i $-$ *Mean distance to closest other cluster*

What if you don’t have Ground Truth?

→ But what if you don’t have a labeled dataset?

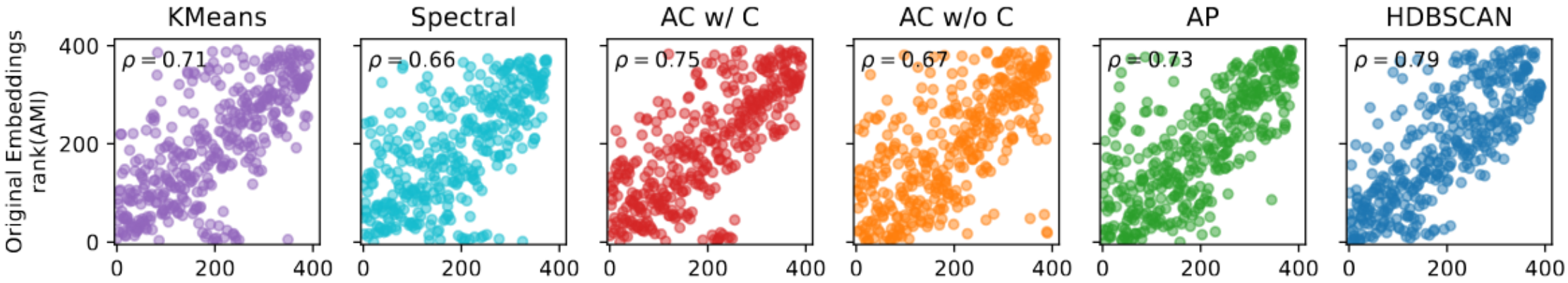
→ Cluster quality can be measured using the *Silhouette score*

→ The Silhouette score is an *intrinsic* measure based on how ”tight” and separated clusters are

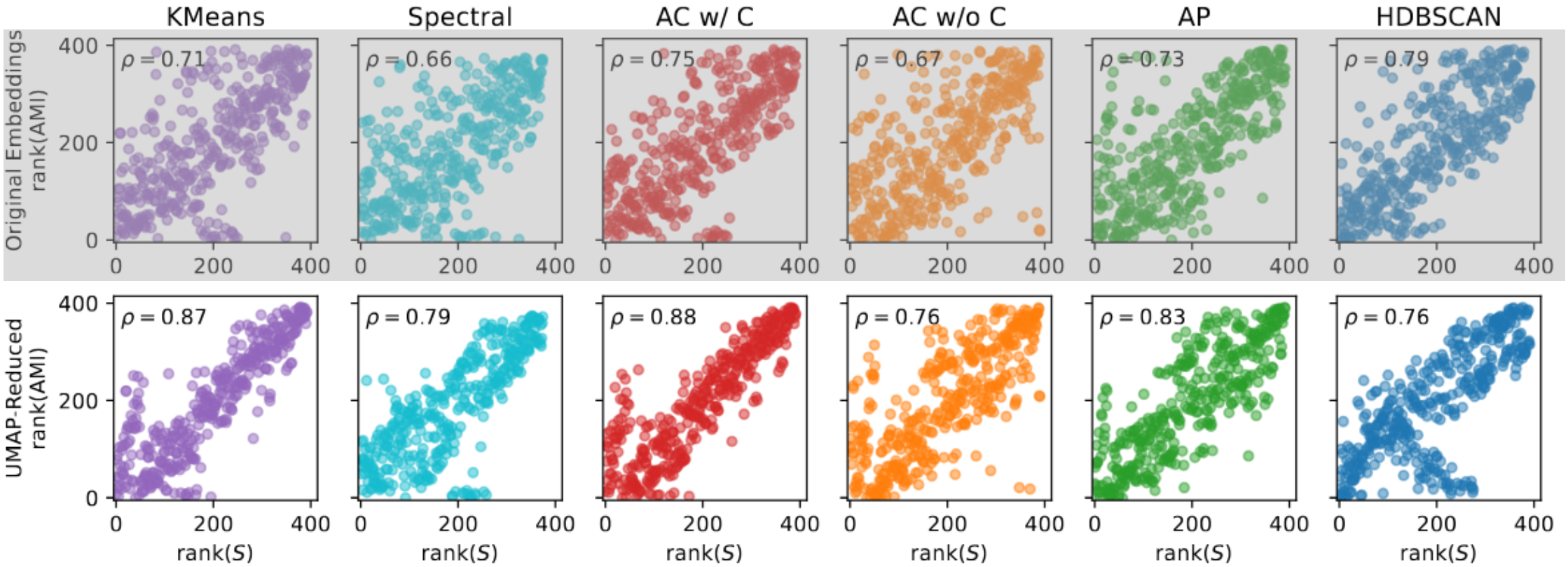
→ Are the GT-based AMI scores and GT-free Silhouette scores correlated?

What if you don’t have Ground Truth?

What if you don’t have Ground Truth?



What if you don't have Ground Truth?



What if you don’t have Ground Truth?

→ **Yes,** Silhouette and AMI scores are correlated!

What if you don’t have Ground Truth?

→ **Yes,** Silhouette and AMI scores are correlated!

→ Allows for proxy-model evalaution on new tasks without any label

→ Much more flexible model selection process compared to

1) kNN probing, which requires stored labeled data points
2) Linear probing, which requires fitting a large linear layer