Self-Supervised Learning

Marina Munkhoeva
Research Scientist, AIRI

HSE, 2025-09-18

00

Outline

Outline

01

Course Logistics

Important info

2nd time the course is taught

so far 12 lectures+seminars planned

2 homeworks + 1 group project

1st homework will be published in October

Schedule: Thursdays 13:00 — 16:00

Telegram chat: ask `@helenlyko` (our TA) to be added

Group Projects

1. Form groups of 3-5 students by start of October!
2. Submit your group member list (TBA)

After you are settled on the project topic, **immediately** start work on Project Proposal

Project proposal specifies the scope of the project and consists of

- Motivation (What problem are you solving? Application or theory?)
- Method (What self-supervised methods are you planning to apply/improve?)
- Intended Experiments (What experiments should be run? How will you evaluate results? What are your baselines?)

Look for relevant datasets and prior research.

Picking a Project

Typically 3 kinds of projects (often a combination of these):

1. Application project. Pick application area of your interest, explore how best to apply one of the learned methods.
2. Algorithmic project. Pick a family of algorithms and develop a new or novel extension of existing algorithm.
3. Theoretical project. Prove some interesting property of new/existing method.

You can always continue work on the project if it seems promising and has potential for publishing (ideally).

Picking a Project

When picking a project

- search related work on `scholar.google.com` / `arxiv.org`, etc
- identify available datasets / benchmarks
- evaluate time and effort needed, e.g. dataset might need involved pre-processing

NB keep in mind time needed for a formal write-up with explanation of methodology and discussion of results.

NB Some projects might continue as bachelor's thesis.

Final Project Notes

Formal write-up and project presentation will be your main deliverables.

Write-ups should be up to 4 pages long (excluding references and **contributions**)

We'll publish format guideline later on for your convenience.

Code - make sure you provide a link to GitHub or zip file.

Grading

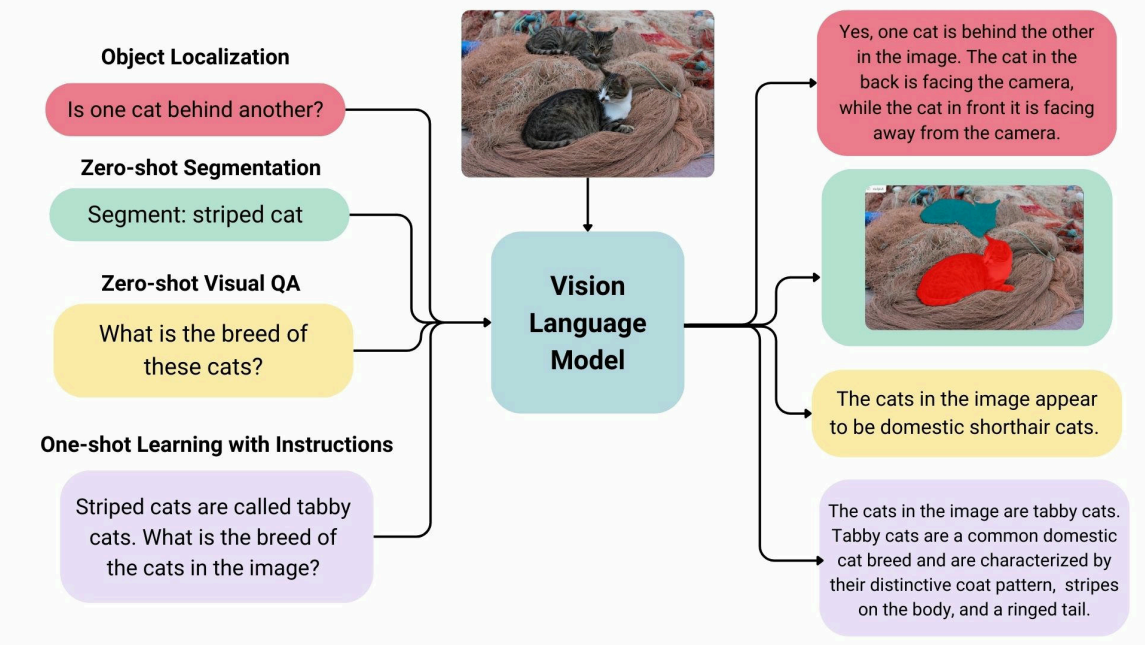
- technical quality of the work
- originality in all aspects (data processing, methods, analysis)
- overall presentation of the ideas and results

02

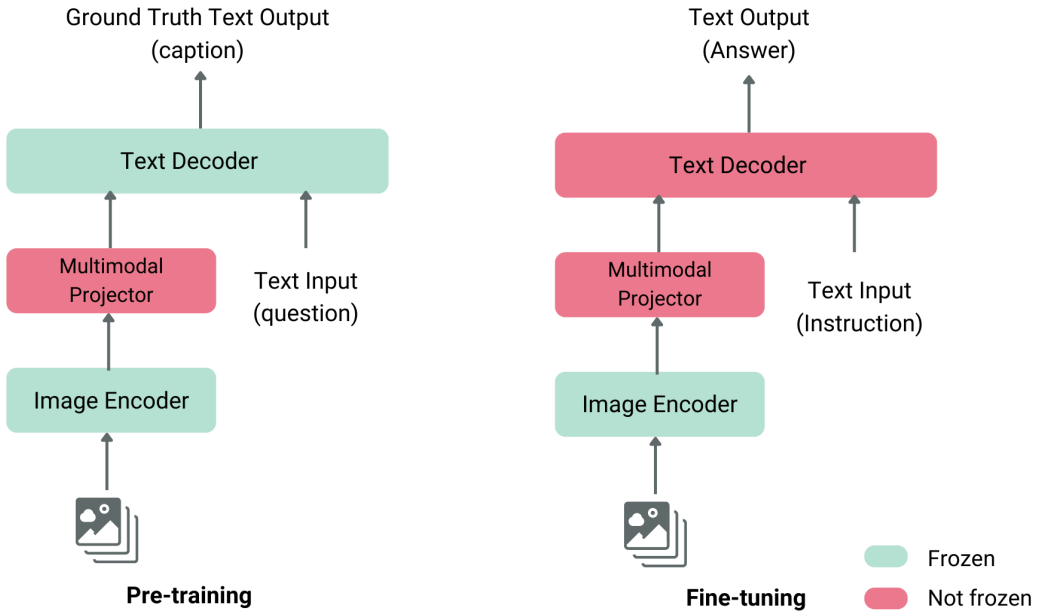
Overview

Intro

What powers multimodal models?



(source)



LLaVA = CLIP image encoder + multimodal projector + Vicuna text decoder.

Problems Travel Downstream

- Spurious correlations from SSL pretraining: If encoder learns "snow = white," multimodal model may caption anything white as "snow." **brittle vision-language grounding**
- Misalignment between text and visual granularity: CLIP-style encoders are trained on noisy, high-level captions. Models capture "chair" and "person" but not "the red ball under the chair." That's why some **VLMs struggle with detailed spatial reasoning**
- Bias inheritance: Web-scale SSL data encodes cultural bias, aesthetic preference, demographic imbalance. **Biases persist** even when plugged into powerful LLMs.
- Hallucinations: When the vision encoder embedding is weak or under-specified, the LLM fills in gaps with its text prior. This causes **multimodal hallucinations**.

Supervised Learning

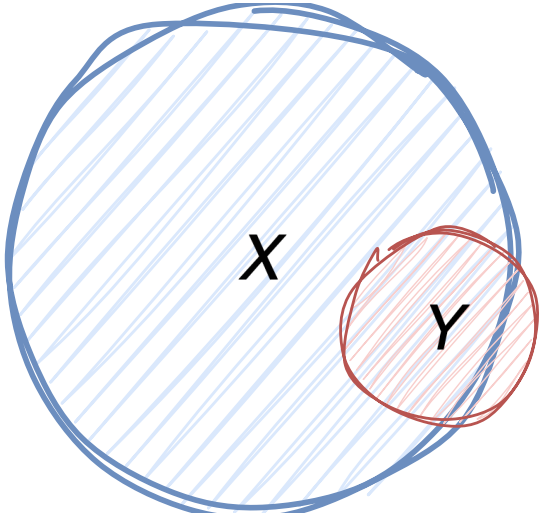
Risk Minimization

Learn $f : X \to Y$ minimizing expected loss over function class $\mathcal{F}$

$$\min_{f \in \mathcal{F}} \mathbb{E}_{x \sim p_X} \mathcal{L}(f(x, \theta), y)$$

Example: X - images, Y - content classes (e.g. a bunny, a bird)

Assumption – training distribution P_X matches target $P_{X'}$



Task Y needs little from X

+	Simple	—	Expensive labelling
+	Theoretical foundations	—	Overfitting risk
+	Clear evaluation	—	Limited generalization

Unsupervised Learning

Discovering intrinsic structure

- no labelled data
- **much harder**

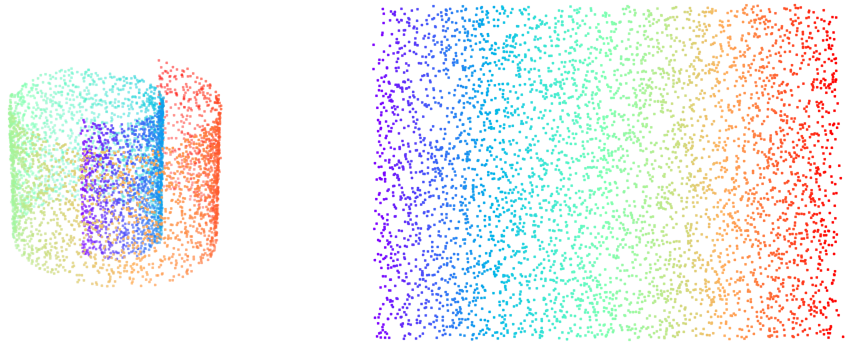
Proxy objectives:

- Clustering
- Dimensionality reduction
- Density Estimation
- Input Space Reconstruction

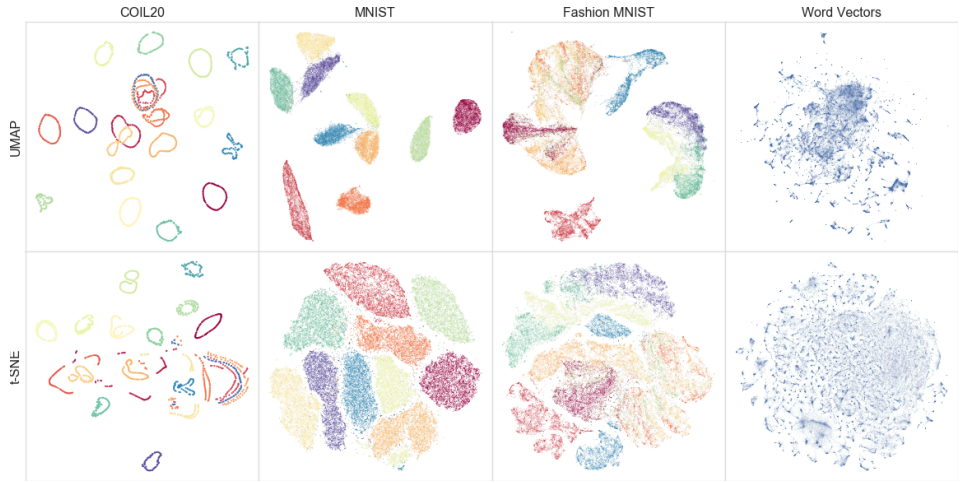
Assumption – Manifold Hypothesis

+ Cheap unlabeled data	— Evaluation challenges
+ Pattern discovery	— Ambiguity in results
+ Data generation	— Computational demands

Swiss Roll Example (2D manifold in 3D)



t-SNE and UMAP on different datasets

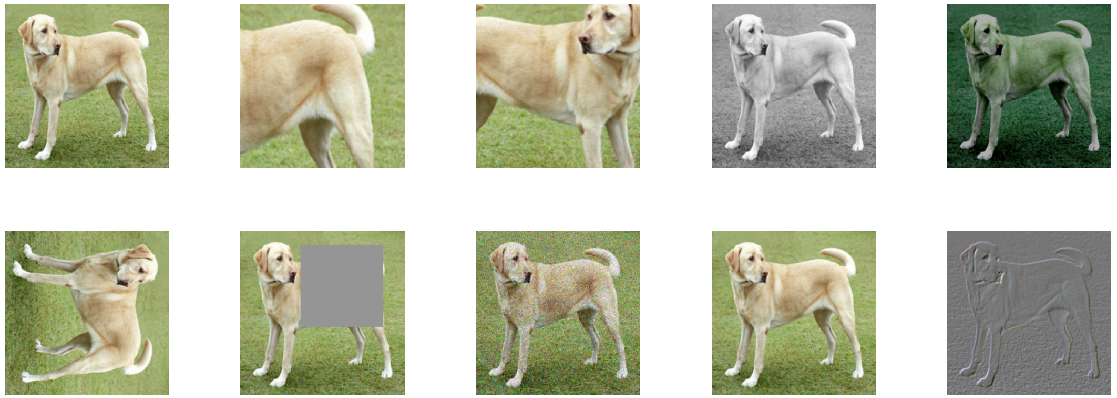


What is Self-Supervised Learning?

No labels? — use cheap domain knowledge to generate **pretext tasks** with **pseudo labels**

Computer Vision

Augment image x to get different views x_i^+:

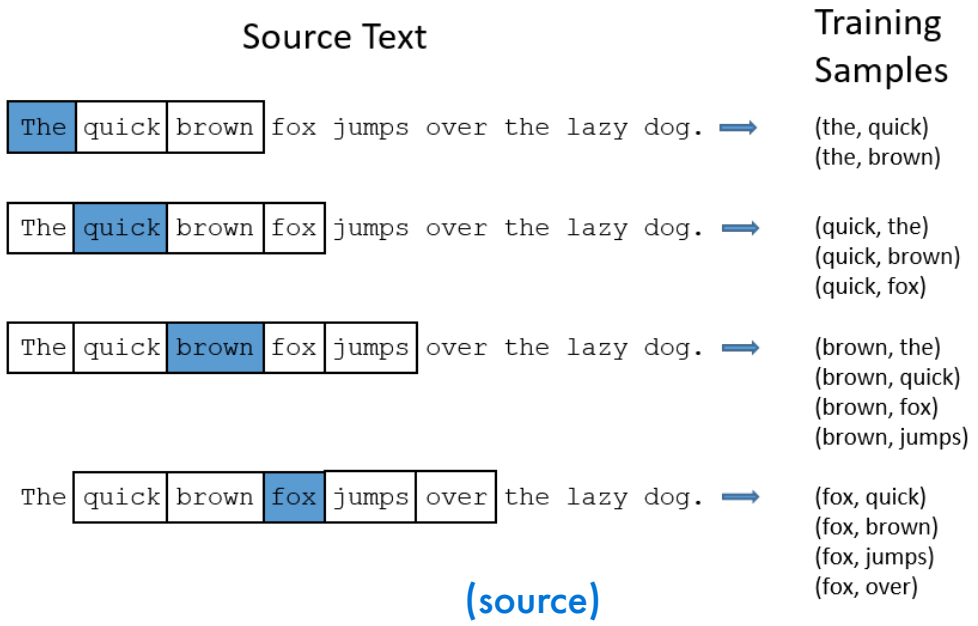


(Chen et al., 2020a)

Sample negatives x_k^-

Attract positives, repel negatives

Natural Language Processing

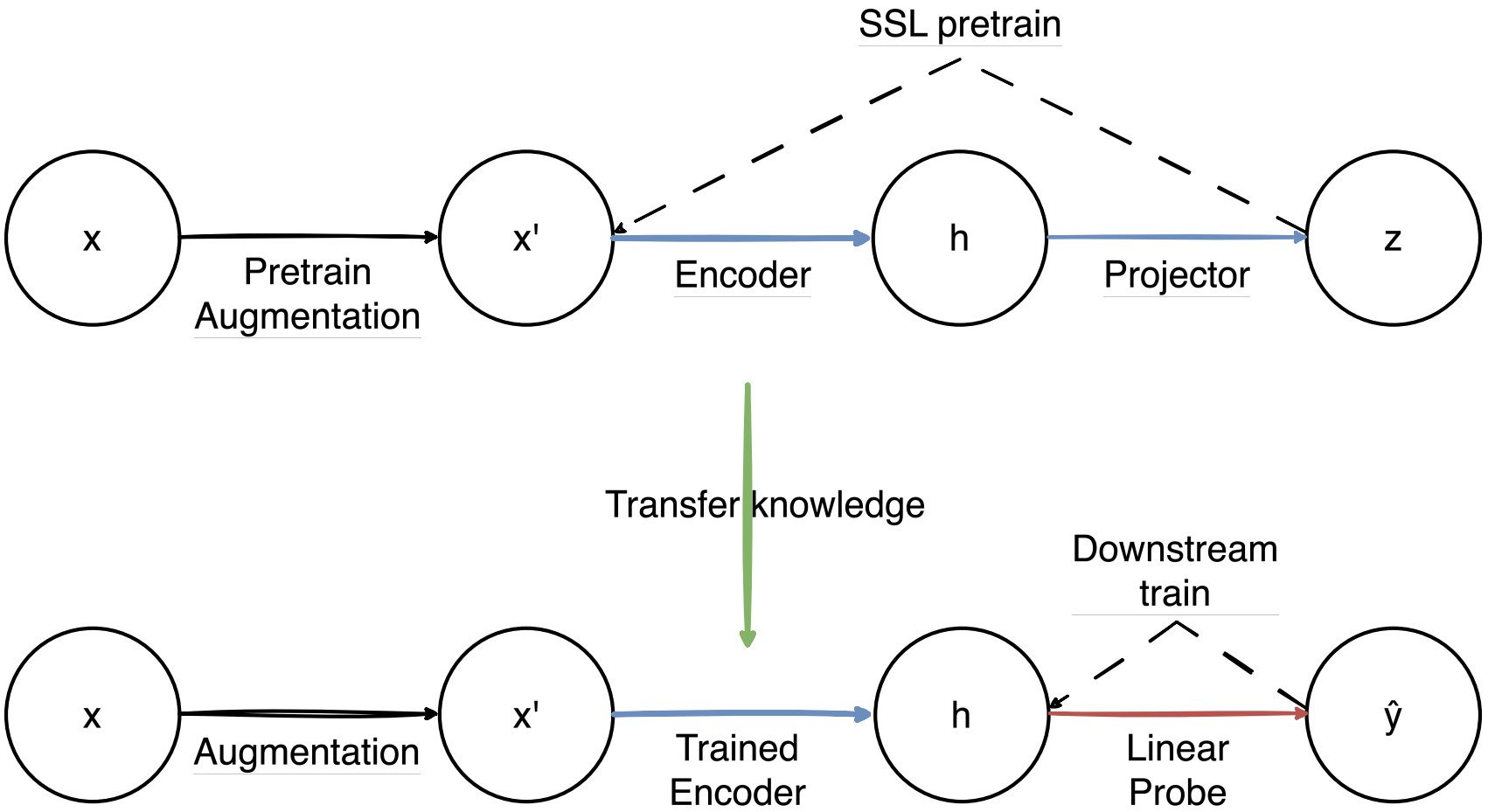


(source)

Sample context as words around position

Learn to predict context

SSL Train Pipeline



Linear Probing

Standard **evaluation protocol** for SSL (flawed, but widespread)

Given a trained encoder $g_\theta : \mathcal{X} \to \mathcal{Z}$, frozen θ

Use linear layer W to map representation $g(x)$ to target space

Find W with SGD (no backprop thru θ!):

$$\min_W \frac{1}{N} \sum_{x,y \in D_{\text{train}}} \text{CrossEntropy}(W(g(x)), y)$$

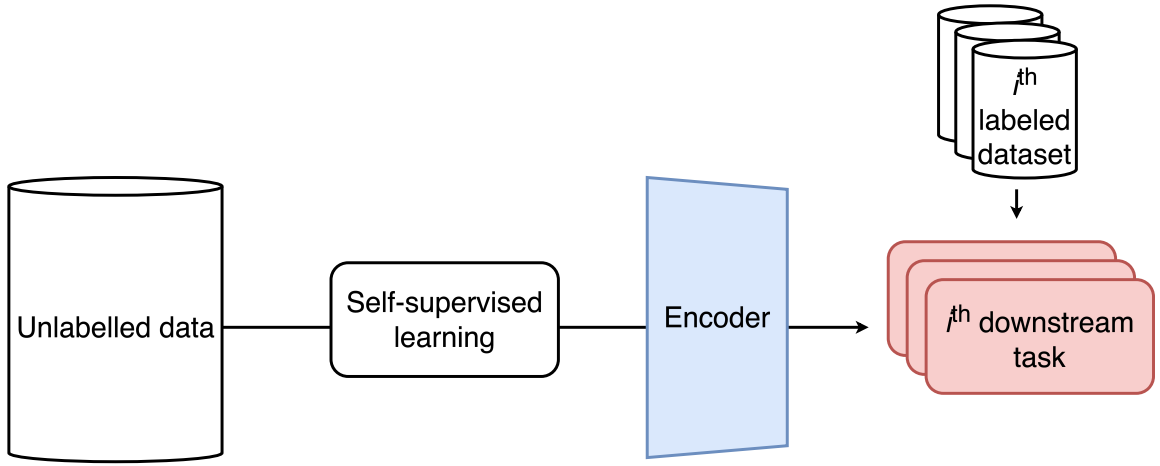
Measure accuracy on test set D_{test}

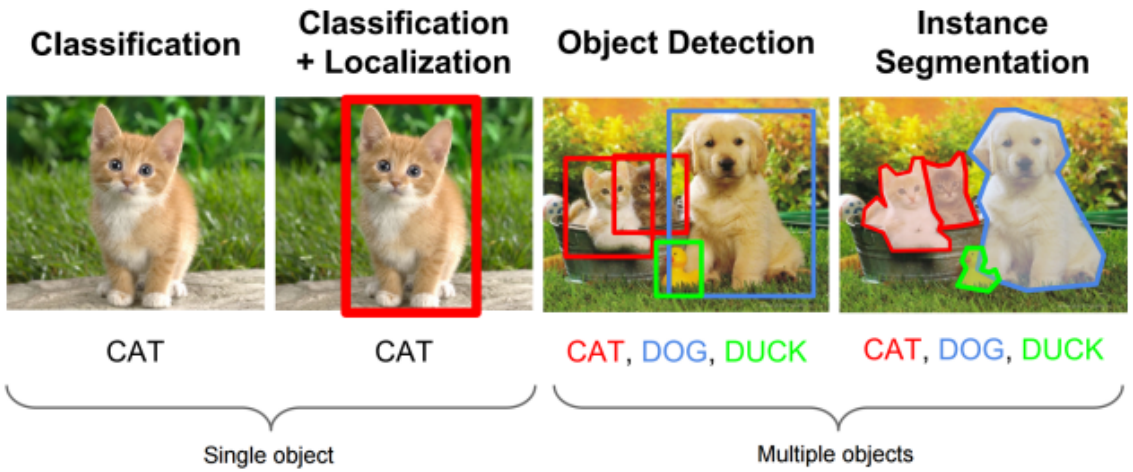
Offline (after pretraining is finished) and online — regularly during pretraining; highly correlates with offline!

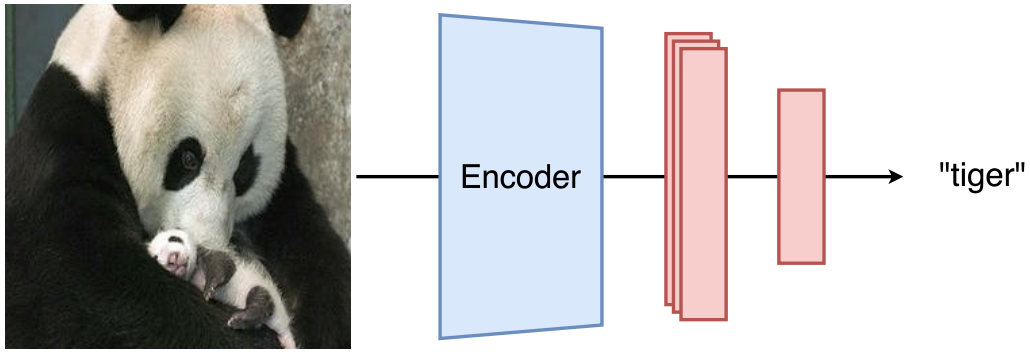
Other common eval option: **k-means**

Overall Pipeline

- Pretrain encoder on unlabeled data with some SSL method
- Learned features are useful in **multiple** downstream tasks







encoder + classification head

Why Self-Supervised Learning?

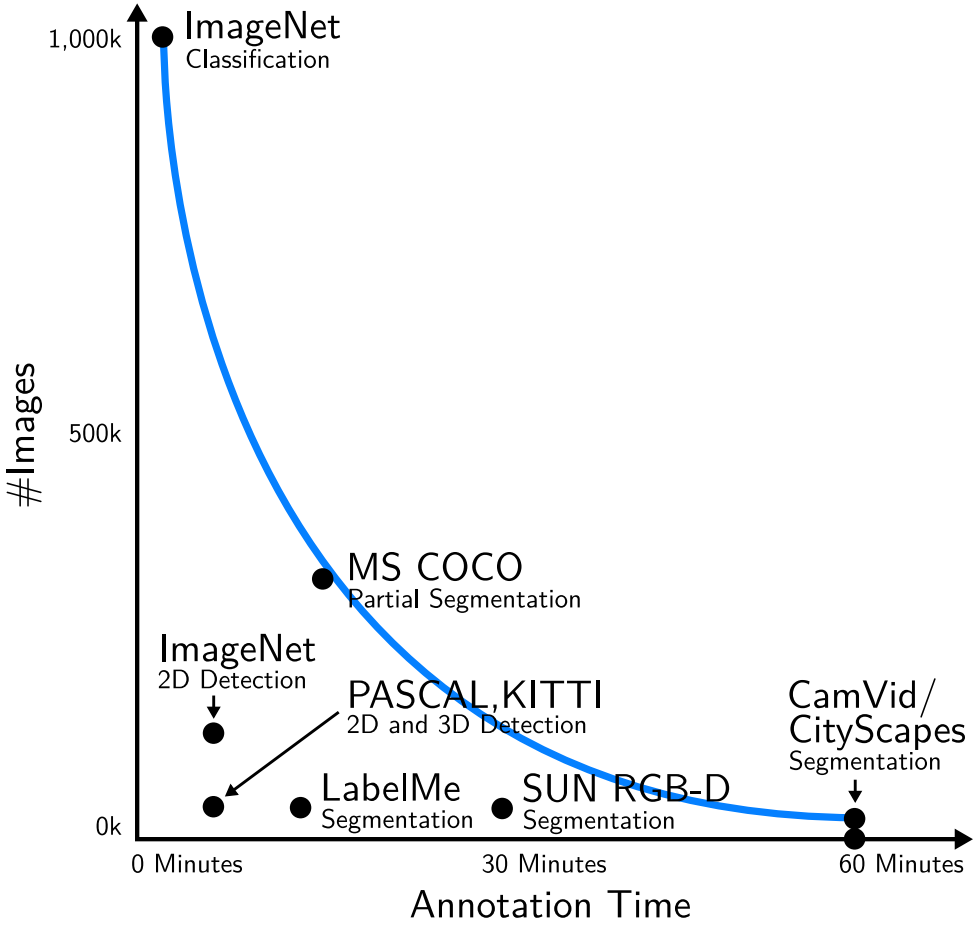
- Labels are task specific
- Labeling is error-prone
- Labelers need training

— Labelled data is expensive

ImageNet: red panda
MTurk: panda

ImageNet: sunglasses
MTurk: + water bottle

See more here >>> (source) <<<



(Xie et al., 2016)

Label: dough; Model: bagel. When does dough become a bagel?

(Vasudevan et al., 2022)

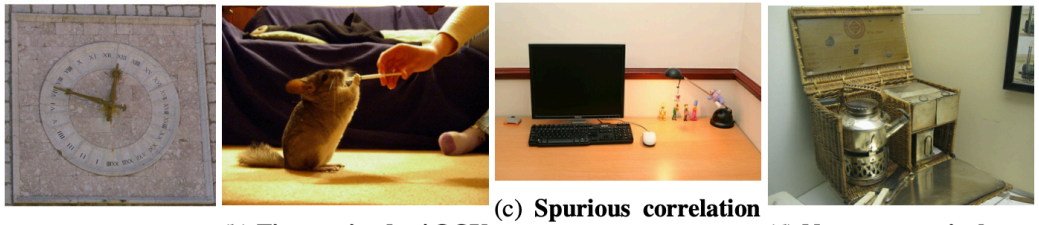
(a) **Major mistake**
Label: dough
Model: jigsaw puzzle

(b) **Minor mistake**
Label: kuvasz
Model: Great Pyrenees

(c) **New multilabel**
Label: tape player
Model: cassette

(d) **Problematic**
Label: bee
Model: fly

Figure 1: **Mistake Severity.** Examples of the two mistake severities (a-b), a correct model prediction where the model identifies a previously missing multi-label (c); and a problematic example (d) where the label (bee) is incorrect (object is a bee-fly, which is a type of fly).



(a) **Fine-grained**
Label: wall clock
Model: sundial

(b) **Fine-grained w/ OOV**
Label: syringe
Model: hamster

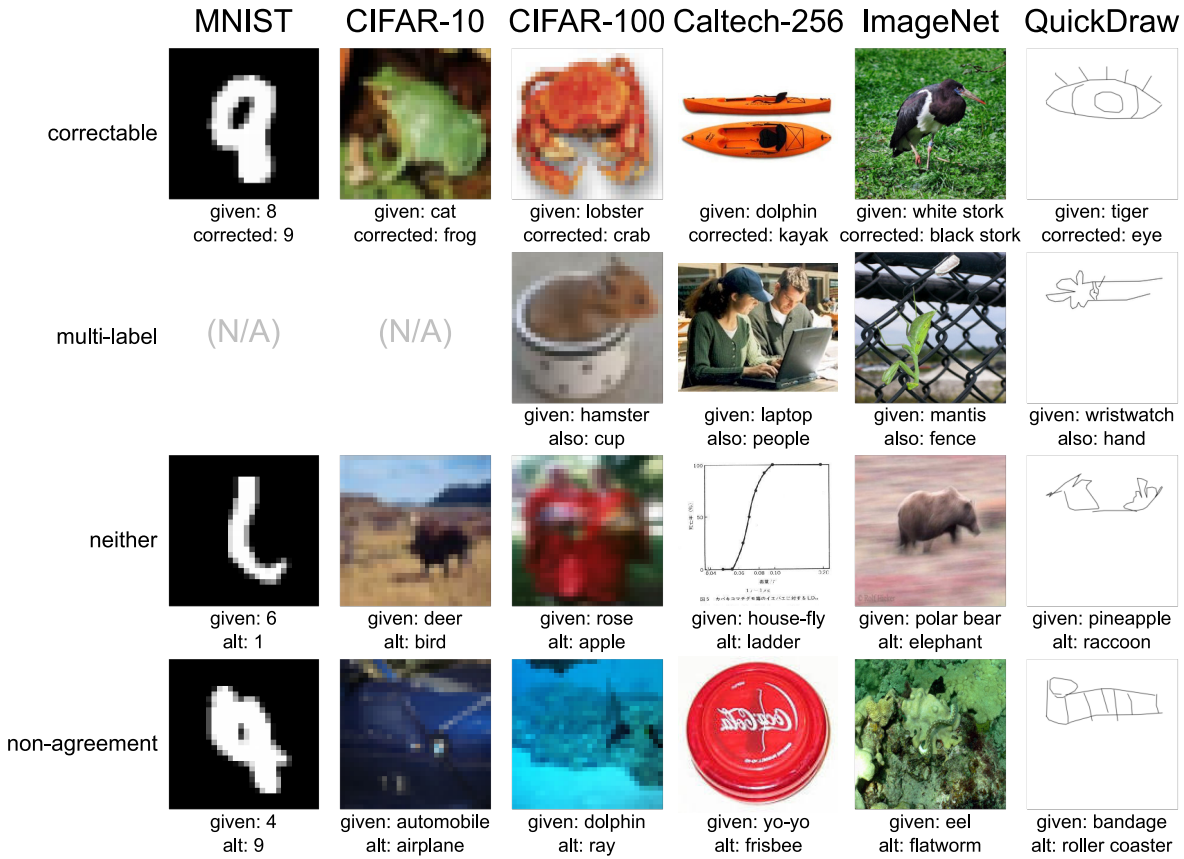
(c) **Spurious correlation**
Label: mouse, desk, monitor, screen
Model: desktop computer

(d) **Non-prototypical**
Label: stove
Model: hamper

Figure 2: **Mistake Category.** Examples of the four mistake categories. In the fine-grained with OOV example, the animal is a chinchilla, which is not an ImageNet class but is visually similar to a hamster, which is an ImageNet class. In the spurious correlation example, the scene contains relevant context for desktop computer, but there is no such object in the image.

(Vasudevan et al., 2022)

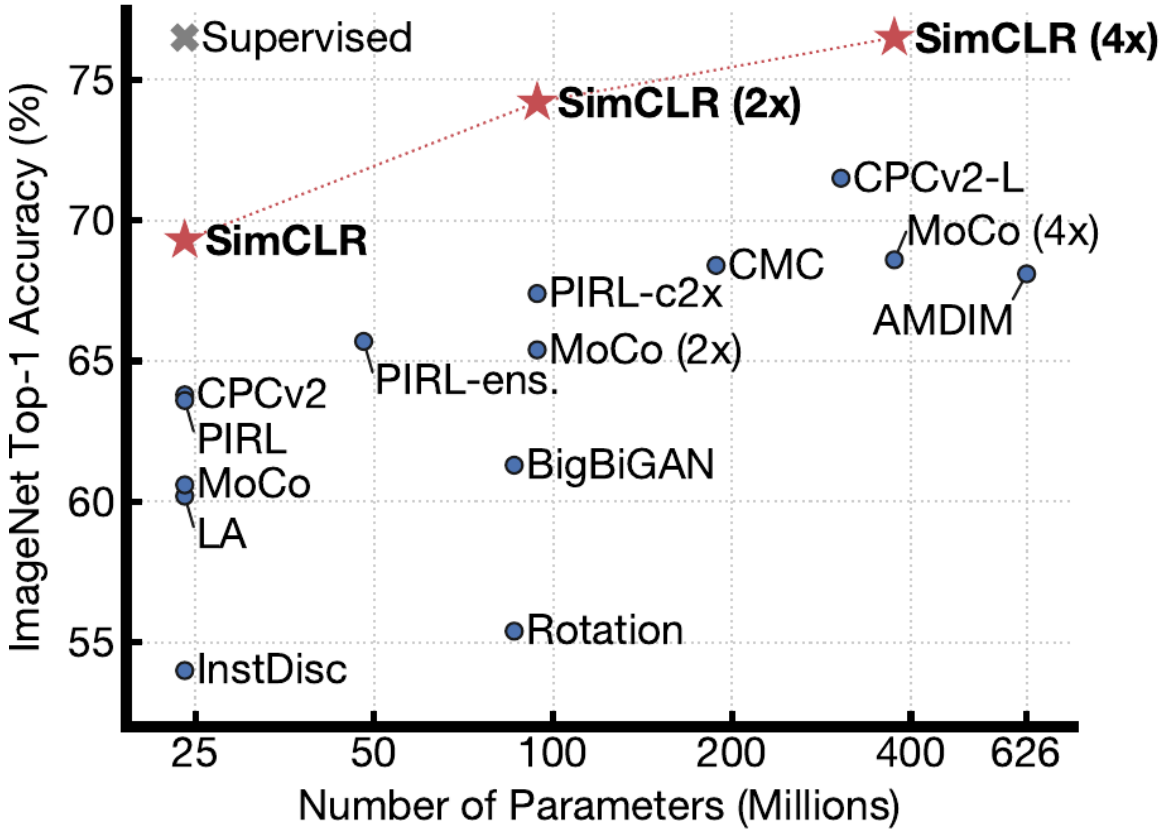
Mislabeling is Pervasive



(Northcutt et al., 2021)

Benefits of SSL

Supervised Performance



(Chen et al., 2020a)

segmentation: supervised vs DINO

(Caron et al., 2021)

Transfer Learning

Storing knowledge from solving one problem and applying it to a different problem.

Pretraining on Imagenet-1k, evaluation on 12 datasets

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear Eval												
SimCLR	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned												
SimCLR	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

(Chen et al., 2020a)

NB Domain adaptation – source and target problems distributions are different (but related): Natural images vs medical images.

Label-Efficient Learning

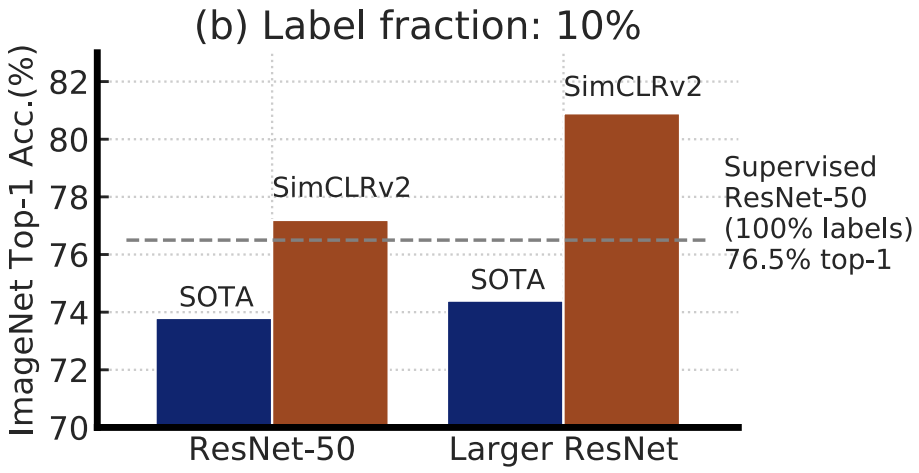
Semi-supervised learning:

- only X% of data is labelled

1. Pretrain encoder on whole dataset
2. Learn probe/finetune (encoder+probe) on labelled part

Further distillation techniques exist

obtain labelled data perf with 10% of labels



SimCLRv2 (Chen et al., 2020b)

Method	Architecture	Label fraction 1%	Label fraction 10%
		Top 5	Top 5
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(*)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

SimCLR (Chen et al., 2020a)

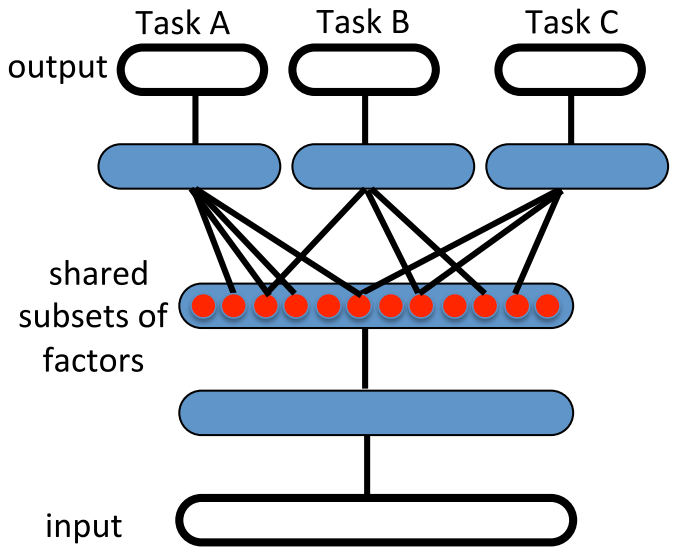
Self-Supervised Learning Summary

General purpose representations
— no single task alignment
— transferrable to many **downstream** tasks

Pretraining technique for data in the wild

Reach supervised performance

Less prone to spurious correlations



(Bengio et al., 2013)

\+ Cheap unlabeled data
\+ Task-agnostic
\+ Scalability

— Computational complexity
— Sensitive to proxy labels
— Limited theoretical understanding

Representation Learning

Supervised approach learns $f : \mathcal{X} \to \mathcal{Y}$
Instead, let's learn $g : \mathcal{X} \to \mathcal{Z}$

Goal: $\mathcal{Z}$ useful for **many** domains/tasks

Learn **simple** classifier/predictor $f_i : \mathcal{Z} \to \mathcal{Y}_i$ on top of g for many tasks Y_i

If learning g requires

- no need for labels
- synthetic labels

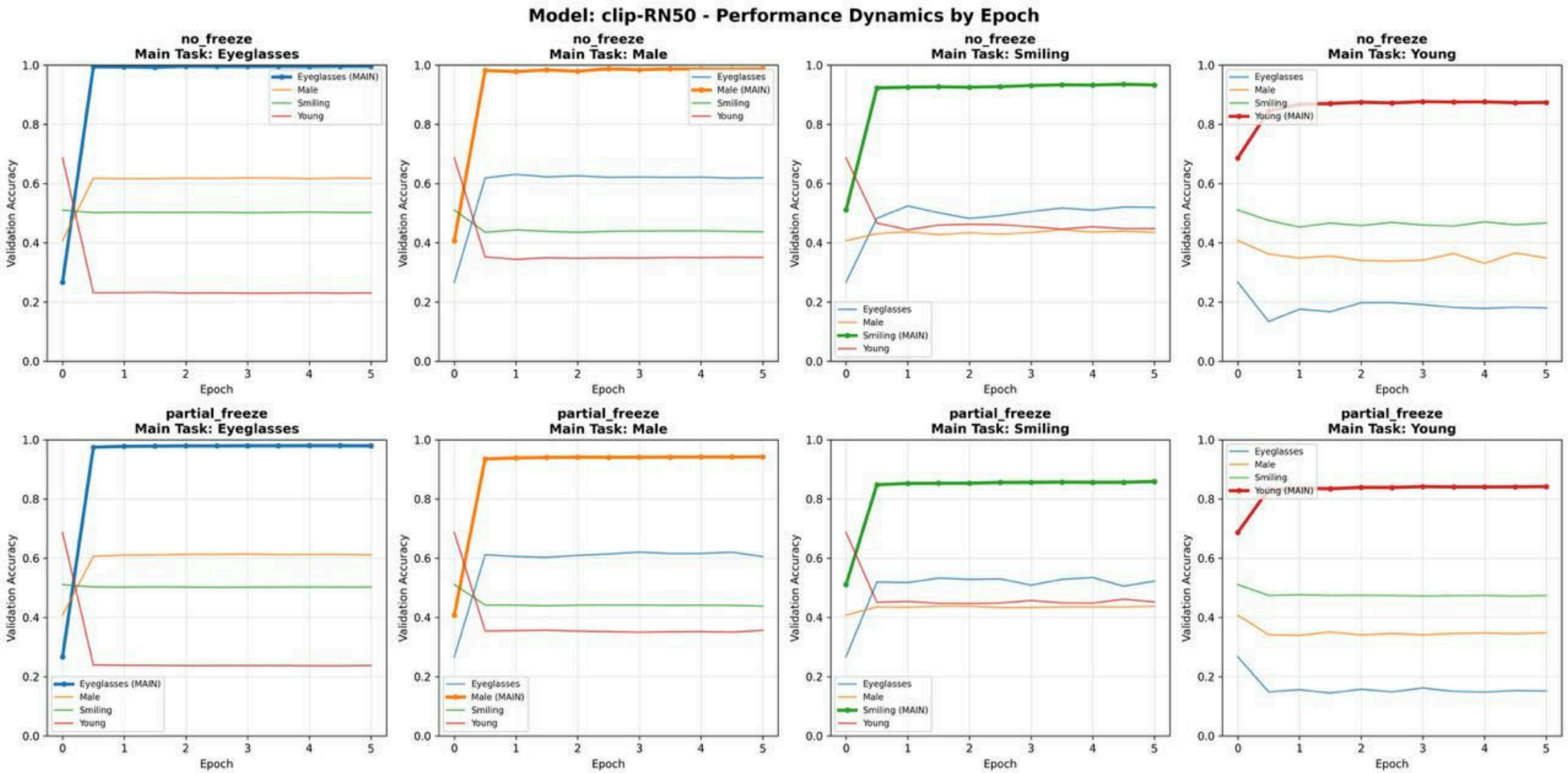
$\Rightarrow$ make use of vast unlabeled data

To maximize representational utility — capture **underlying** factors of variation

Identifying explanatory factors is a challenge! (spurious features)

(Izmailov et al., 2022)

label correlations



What makes representations good?

Smoothness: if $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$

What makes representations good?

Smoothness: if $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$

Invariances/Equivariance: informally, $x_2 = t(x_1) \Rightarrow g(x_2) = g(x_1)$ (invariance) or $g(x_2) = \hat{t}(g(x_1))$ (equivariance)

What makes representations good?

Smoothness: if $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$

Invariances/Equivariance: informally, $x_2 = t(x_1) \Rightarrow g(x_2) = g(x_1)$ (invariance) or $g(x_2) = \hat{t}(g(x_1))$ (equivariance)

~~Task~~ **Domain specific**: invariance under **shared domain** transformations, e.g. color jitter

What makes representations good?

Smoothness: if $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$

Invariances/Equivariance: informally, $x_2 = t(x_1) \Rightarrow g(x_2) = g(x_1)$ (invariance) or $g(x_2) = \hat{t}(g(x_1))$ (equivariance)

~~**Task**~~ **Domain specific**: invariance under **shared domain** transformations, e.g. color jitter

Natural clustering: representation should reflect categorical nature of data

What makes representations good?

Smoothness: if $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$

Invariances/Equivariance: informally, $x_2 = t(x_1) \Rightarrow g(x_2) = g(x_1)$ (invariance) or $g(x_2) = \hat{t}(g(x_1))$ (equivariance)

~~**Task**~~ **Domain specific**: invariance under **shared domain** transformations, e.g. color jitter

Natural clustering: representation should reflect categorical nature of data

Multiple explanatory factors: diverse tasks require diverse features

What makes representations good?

Smoothness: if $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$

Invariances/Equivariance: informally, $x_2 = t(x_1) \Rightarrow g(x_2) = g(x_1)$ (invariance) or $g(x_2) = \hat{t}(g(x_1))$ (equivariance)

~~**Task**~~ **Domain specific**: invariance under **shared domain** transformations, e.g. color jitter

Natural clustering: representation should reflect categorical nature of data

Multiple explanatory factors: diverse tasks require diverse features

Disentangle underlying factors: e.g. z_i — lighting, z_j — color tone, z_k — texture

(Hudson et al., 2024)

What makes representations good?

Smoothness: informally, if $x_1 \approx x_2$, then $g(x_1) \approx g(x_2)$

Invariances/Equivariance: informally, $x_2 = t(x_1) \Rightarrow g(x_2) = g(x_1)$ **(invariance)** or $g(x_2) = \hat{t}(g(x_1))$ **(equivariance)**

~~**Task**~~ **Domain specific**: invariance under **shared domain** transformations, e.g. color jitter

Natural clustering: representation should reflect categorical nature of data

Multiple explanatory factors: diverse tasks require diverse features

Disentangle underlying factors: z_i — fringe/no fringe, z_j — age, z_k — smile, etc

Hierarchical explanatory factors: z_i — image style, z_j — specific object

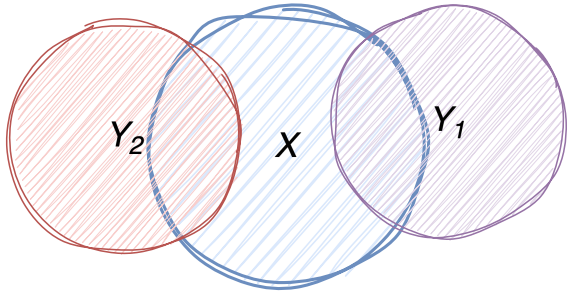
based on representation learning survey from 2013 (Bengio et al., 2013)

Intuitive Information Content

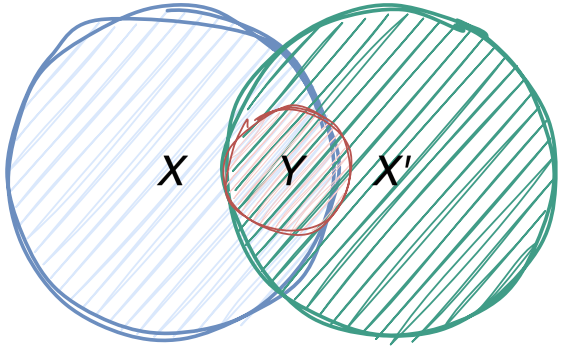
Information theory often used to explain SSL

When does SSL work?

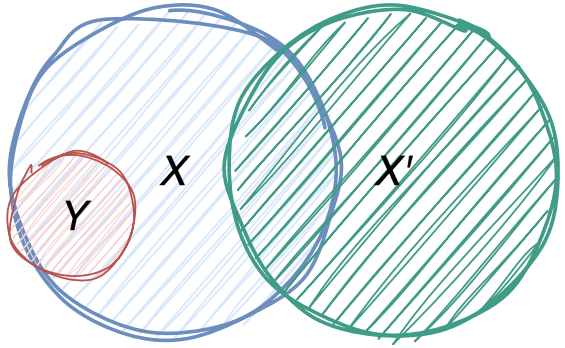
Tasks Y_1 and Y_2 may care about different features
— one's **style** is other's **content**



Data augmentation **hits the spot** for task Y



Data augmentation/second view **irrelevant** for task Y



03

Bibliography

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798–1828.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. Proceedings of the IEEE/CVF International Conference on Computer Vision, 9650–9660.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. International Conference on Machine Learning, 1597–1607.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. E. (2020b). Big self-supervised models are strong semi-supervised learners. Advances in Neural Information Processing Systems, 33, 22243–22255.

Hudson, D. A., Zoran, D., Malinowski, M., Lampinen, A. K., Jaegle, A., McClelland, J. L., Matthey, L., Hill, F., & Lerchner, A. (2024). Soda: Bottleneck diffusion models for

representation learning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 23115–23127.

Izmailov, P., Kirichenko, P., Gruver, N., & Wilson, A. G. (2022). On feature learning in the presence of spurious correlations. Advances in Neural Information Processing Systems, 35, 38516–38532.

Northcutt, C. G., Athalye, A., & Mueller, J. (2021). Pervasive label errors in test sets destabilize machine learning benchmarks. Arxiv Preprint Arxiv:2103.14749.

Vasudevan, V., Caine, B., Gontijo Lopes, R., Fridovich-Keil, S., & Roelofs, R. (2022). When does dough become a bagel? analyzing the remaining mistakes on imagenet. Advances in Neural Information Processing Systems, 35, 6720–6734.

Xie, J., Kiefel, M., Sun, M.-T., & Geiger, A. (2016). Semantic instance annotation of street scenes by 3d to 2d label transfer. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3688–3697.

Thank you!