Self-supervised and Representation Learning

Computer Vision (CSCI 5520G)

Faisal Z. Qureshi

http://vclab.science.ontariotechu.ca

Week 8 at a glance

- **The labelling bottleneck**
- **What makes a representation good, and how we measure it**
- **Pretext tasks** — and why they stalled
- **Contrastive learning** — InfoNCE, SimCLR, MoCo
- **Avoiding collapse without negatives** — BYOL, SimSiam, DINO
- **Masked image modelling** — MAE
- **What self-supervision does not fix**

Reading: SimCLR (Chen et al., ICML 2020), MAE (He et al., CVPR 2022), DINO (Caron et al., ICCV 2021).

Related notes: t-SNE, autoencoders.

The bottleneck

Everything so far has been supervised. Supervision is expensive.

The bottleneck

Everything so far has been supervised. Supervision is expensive.

- ImageNet: 14 million images, hand-labelled over years
- Segmentation masks: minutes per image, not seconds
- Medical imaging: the annotator must be a radiologist
- Remote sensing, microscopy, industrial inspection: the same story

The bottleneck

Everything so far has been supervised. Supervision is expensive.

- ImageNet: 14 million images, hand-labelled over years
- Segmentation masks: minutes per image, not seconds
- Medical imaging: the annotator must be a radiologist
- Remote sensing, microscopy, industrial inspection: the same story

Meanwhile there are billions of unlabelled images.

The bottleneck

Everything so far has been supervised. Supervision is expensive.

- ImageNet: 14 million images, hand-labelled over years
- Segmentation masks: minutes per image, not seconds
- Medical imaging: the annotator must be a radiologist
- Remote sensing, microscopy, industrial inspection: the same story

Meanwhile there are billions of unlabelled images.

The question of this lecture: can we learn useful visual representations without labels, and then spend our small labelling budget only on the final task?

What is a "good" representation?

We want a function f mapping images to vectors such that:

What is a "good" representation?

We want a function f mapping images to vectors such that:

- **Linearly separable.** Downstream tasks need only a simple head.
- **Transferable.** Useful for tasks not known at training time.
- **Invariant** to nuisances — lighting, crop, colour jitter.
- **Equivariant** where it matters — position, for detection.

What is a "good" representation?

We want a function f mapping images to vectors such that:

- **Linearly separable.** Downstream tasks need only a simple head.
- **Transferable.** Useful for tasks not known at training time.
- **Invariant** to nuisances — lighting, crop, colour jitter.
- **Equivariant** where it matters — position, for detection.

Notice the tension in the last two. Invariance to translation helps classification and *hurts* localisation.

What is a "good" representation?

We want a function f mapping images to vectors such that:

- **Linearly separable.** Downstream tasks need only a simple head.
- **Transferable.** Useful for tasks not known at training time.
- **Invariant** to nuisances — lighting, crop, colour jitter.
- **Equivariant** where it matters — position, for detection.

Notice the tension in the last two. Invariance to translation helps classification and *hurts* localisation.

There is no task-free notion of a good representation. Every method encodes an opinion about what should be discarded.

How we measure it

Protocol matters more than most papers admit.

How we measure it

Protocol matters more than most papers admit.

- **Linear probe.** Freeze f, train a linear classifier. Measures linear separability.
- **Fine-tuning.** Train everything. Measures initialisation quality — and can hide a poor representation.
- **k-NN.** No training at all. Measures metric structure directly.
- **Few-shot / low-data.** The regime self-supervision is actually *for*.

How we measure it

Protocol matters more than most papers admit.

- **Linear probe.** Freeze f, train a linear classifier. Measures linear separability.
- **Fine-tuning.** Train everything. Measures initialisation quality — and can hide a poor representation.
- **k-NN.** No training at all. Measures metric structure directly.
- **Few-shot / low-data.** The regime self-supervision is actually *for*.

Methods reorder under different protocols. MAE is unremarkable under a linear probe and excellent under fine-tuning; contrastive methods often show the opposite pattern.

How we measure it

Protocol matters more than most papers admit.

- **Linear probe.** Freeze f, train a linear classifier. Measures linear separability.
- **Fine-tuning.** Train everything. Measures initialisation quality — and can hide a poor representation.
- **k-NN.** No training at all. Measures metric structure directly.
- **Few-shot / low-data.** The regime self-supervision is actually *for*.

Methods reorder under different protocols. MAE is unremarkable under a linear probe and excellent under fine-tuning; contrastive methods often show the opposite pattern.

When you read one of these papers, find the protocol before you read the table.

Pretext tasks: the first idea

Invent a task whose labels come free from the image itself, and hope the features learned to solve it are useful.

Pretext tasks: the first idea

Invent a task whose labels come free from the image itself, and hope the features learned to solve it are useful.

- **Rotation.** Predict which of $0°, 90°, 180°, 270°$.
- **Jigsaw.** Shuffle patches, predict the permutation.
- **Colourisation.** Predict chrominance from luminance.
- **Inpainting.** Reconstruct a masked-out region.
- **Relative patch location.** Given two patches, predict their arrangement.

Pretext tasks: the first idea

Invent a task whose labels come free from the image itself, and hope the features learned to solve it are useful.

- **Rotation.** Predict which of $0°, 90°, 180°, 270°$.
- **Jigsaw.** Shuffle patches, predict the permutation.
- **Colourisation.** Predict chrominance from luminance.
- **Inpainting.** Reconstruct a masked-out region.
- **Relative patch location.** Given two patches, predict their arrangement.

These worked — somewhat. They consistently trailed supervised pretraining.

Why pretext tasks stalled

The features solve *the pretext task*, and the pretext task is a proxy we chose.

Why pretext tasks stalled

The features solve *the pretext task*, and the pretext task is a proxy we chose.

- Rotation prediction can be solved by finding the sky. That is a real cue, and a shallow one.
- Colourisation rewards texture statistics over object identity.
- Jigsaw can be won on low-level edge continuity at patch boundaries — so much so that papers had to add gaps between patches to prevent it.

Why pretext tasks stalled

The features solve *the pretext task*, and the pretext task is a proxy we chose.

- Rotation prediction can be solved by finding the sky. That is a real cue, and a shallow one.
- Colourisation rewards texture statistics over object identity.
- Jigsaw can be won on low-level edge continuity at patch boundaries — so much so that papers had to add gaps between patches to prevent it.

The general failure mode: shortcut learning. The network finds the cheapest solution to the task you posed, which is rarely the one you meant.

Why pretext tasks stalled

The features solve *the pretext task*, and the pretext task is a proxy we chose.

- Rotation prediction can be solved by finding the sky. That is a real cue, and a shallow one.
- Colourisation rewards texture statistics over object identity.
- Jigsaw can be won on low-level edge continuity at patch boundaries — so much so that papers had to add gaps between patches to prevent it.

The general failure mode: shortcut learning. The network finds the cheapest solution to the task you posed, which is rarely the one you meant.

This is worth generalising. Any time you design a loss, ask what the cheapest way to minimise it is.

Contrastive learning

Change the objective. Rather than solving an invented task, learn a metric: **an image should be close to a transformed version of itself, and far from other images.**

Contrastive learning

Change the objective. Rather than solving an invented task, learn a metric: **an image should be close to a transformed version of itself, and far from other images.**

Given an anchor $\mathbf{x}$, a positive $\mathbf{x}^+$ (an augmentation of $\mathbf{x}$), and negatives $\mathbf{x}_1^- \ldots \mathbf{x}_K^-$, the **InfoNCE** loss is

$$\mathcal{L} = -\log \frac{\exp\big(\mathsf{sim}(\mathbf{z}, \mathbf{z}^+)/\tau\big)}{\exp(\mathsf{sim}(\mathbf{z}, \mathbf{z}^+)/\tau) + \sum_k \exp\Big(\mathsf{sim}(\mathbf{z}, \mathbf{z}_k^-)/\tau\Big)}$$

Contrastive learning

Change the objective. Rather than solving an invented task, learn a metric: **an image should be close to a transformed version of itself, and far from other images.**

Given an anchor $\mathbf{x}$, a positive $\mathbf{x}^+$ (an augmentation of $\mathbf{x}$), and negatives $\mathbf{x}_1^- \ldots \mathbf{x}_K^-$, the **InfoNCE** loss is

$$\mathcal{L} = -\log \frac{\exp\big(\mathsf{sim}(\mathbf{z}, \mathbf{z}^+)/\tau\big)}{\exp(\mathsf{sim}(\mathbf{z}, \mathbf{z}^+)/\tau) + \sum_k \exp\Big(\mathsf{sim}(\mathbf{z}, \mathbf{z}_k^-)/\tau\Big)}$$

where $\mathbf{z} = f(\mathbf{x})$, sim is cosine similarity, and τ is a temperature.

Contrastive learning

Change the objective. Rather than solving an invented task, learn a metric: **an image should be close to a transformed version of itself, and far from other images.**

Given an anchor $\mathbf{x}$, a positive $\mathbf{x}^+$ (an augmentation of $\mathbf{x}$), and negatives $\mathbf{x}_1^- \ldots \mathbf{x}_K^-$, the **InfoNCE** loss is

$$\mathcal{L} = -\log \frac{\exp\big(\mathsf{sim}(\mathbf{z}, \mathbf{z}^+)/\tau\big)}{\exp(\mathsf{sim}(\mathbf{z}, \mathbf{z}^+)/\tau) + \sum_k \exp\Big(\mathsf{sim}(\mathbf{z}, \mathbf{z}_k^-)/\tau\Big)}$$

where $\mathbf{z} = f(\mathbf{x})$, sim is cosine similarity, and τ is a temperature.

This is a $(K{+}1)$-way classification problem: *which of these is the positive?*

Augmentation is the design

The positive pair defines what the representation must ignore.

Augmentation is the design

The positive pair defines what the representation must ignore.

Choose random crop and colour jitter, and you declare: *crop and colour do not change identity*.

Augmentation is the design

The positive pair defines what the representation must ignore.

Choose random crop and colour jitter, and you declare: *crop and colour do not change identity*.

That is a strong claim, and sometimes false:

- Colour jitter destroys the representation for a bird-species task where plumage colour *is* the label
- Aggressive cropping can produce two views containing different objects

Augmentation is the design

The positive pair defines what the representation must ignore.

Choose random crop and colour jitter, and you declare: *crop and colour do not change identity*.

That is a strong claim, and sometimes false:

- Colour jitter destroys the representation for a bird-species task where plumage colour *is* the label
- Aggressive cropping can produce two views containing different objects

The augmentation set is not a detail — it is where the human supervision went. We did not remove the prior; we moved it from labels into transformations.

SimCLR

The clean, minimal recipe:

SimCLR

The clean, minimal recipe:

- Two augmented views per image; other images in the batch are the negatives
- A ResNet encoder f
- A small MLP **projection head** g; the loss is applied to $g(f(\mathbf{x}))$, but the representation used downstream is $f(\mathbf{x})$
- Large batches — 4096 — to supply enough negatives
- Long training, LARS optimiser

SimCLR

The clean, minimal recipe:

- Two augmented views per image; other images in the batch are the negatives
- A ResNet encoder f
- A small MLP **projection head** g; the loss is applied to $g(f(\mathbf{x}))$, but the representation used downstream is $f(\mathbf{x})$
- Large batches — 4096 — to supply enough negatives
- Long training, LARS optimiser

The projection head finding is the interesting one. The layer just before the loss is *worse* for transfer than the layer before it. The loss demands invariances that discard information useful elsewhere; the head absorbs that damage.

The SimCLR pipeline

[Figure placeholder]

Two augmented views of one image, a shared encoder, a projection head, and InfoNCE computed over the batch — every other image is a negative

MoCo: decoupling negatives from batch size

Needing 4096-image batches ties research to large-scale hardware.

MoCo: decoupling negatives from batch size

Needing 4096-image batches ties research to large-scale hardware.

Momentum contrast stores negatives in a queue of past encoded batches, so the number of negatives is independent of batch size.

MoCo: decoupling negatives from batch size

Needing 4096-image batches ties research to large-scale hardware.

Momentum contrast stores negatives in a queue of past encoded batches, so the number of negatives is independent of batch size.

Problem: those stored keys were produced by an older, now-stale encoder.

MoCo: decoupling negatives from batch size

Needing 4096-image batches ties research to large-scale hardware.

Momentum contrast stores negatives in a queue of past encoded batches, so the number of negatives is independent of batch size.

Problem: those stored keys were produced by an older, now-stale encoder.

Solution: encode keys with a **momentum encoder** whose weights are an exponential moving average of the query encoder,

$$\theta_k \leftarrow m\,\theta_k + (1-m)\,\theta_q, \qquad m = 0.999$$

so the key encoder drifts slowly and the queue stays approximately consistent.

MoCo: decoupling negatives from batch size

Needing 4096-image batches ties research to large-scale hardware.

Momentum contrast stores negatives in a queue of past encoded batches, so the number of negatives is independent of batch size.

Problem: those stored keys were produced by an older, now-stale encoder.

Solution: encode keys with a **momentum encoder** whose weights are an exponential moving average of the query encoder,

$$\theta_k \leftarrow m\,\theta_k + (1-m)\,\theta_q, \qquad m = 0.999$$

so the key encoder drifts slowly and the queue stays approximately consistent.

Result: comparable quality on ordinary hardware. Worth noting how often a scaling constraint, not an idea, is what a paper actually removes.

The collapse problem

Why do we need negatives at all?

The collapse problem

Why do we need negatives at all?

Without them, the objective "make two views of the same image similar" has a trivial global optimum:

$$f(\mathbf{x}) = \text{constant}, \qquad \forall \mathbf{x}$$

The collapse problem

Why do we need negatives at all?

Without them, the objective "make two views of the same image similar" has a trivial global optimum:

$$f(\mathbf{x}) = \text{constant}, \qquad \forall \mathbf{x}$$

Every pair is then perfectly similar. Loss zero. Representation worthless.

The collapse problem

Why do we need negatives at all?

Without them, the objective "make two views of the same image similar" has a trivial global optimum:

$$f(\mathbf{x}) = \text{constant}, \qquad \forall \mathbf{x}$$

Every pair is then perfectly similar. Loss zero. Representation worthless.

Negatives prevent this by requiring different images to be *dissimilar* — a repulsive term that keeps the embedding spread out.

The collapse problem

Why do we need negatives at all?

Without them, the objective "make two views of the same image similar" has a trivial global optimum:

$$f(\mathbf{x}) = \text{constant}, \qquad \forall \mathbf{x}$$

Every pair is then perfectly similar. Loss zero. Representation worthless.

Negatives prevent this by requiring different images to be *dissimilar* — a repulsive term that keeps the embedding spread out.

But negatives are expensive, and they are noisy: two images of the same class, sampled as negatives, are pushed apart for no good reason.

BYOL and SimSiam: no negatives

Both avoid collapse without any repulsive term.

BYOL and SimSiam: no negatives

Both avoid collapse without any repulsive term.

BYOL. An *online* network predicts the output of a *target* network. The target is an EMA of the online network. Only the online network receives gradients.

BYOL and SimSiam: no negatives

Both avoid collapse without any repulsive term.

BYOL. An *online* network predicts the output of a *target* network. The target is an EMA of the online network. Only the online network receives gradients.

SimSiam. Shows the EMA is not essential. What matters is:

- an asymmetric **predictor** MLP on one branch, and
- a **stop-gradient** on the other

BYOL and SimSiam: no negatives

Both avoid collapse without any repulsive term.

BYOL. An *online* network predicts the output of a *target* network. The target is an EMA of the online network. Only the online network receives gradients.

SimSiam. Shows the EMA is not essential. What matters is:

- an asymmetric **predictor** MLP on one branch, and
- a **stop-gradient** on the other

Remove either and the network collapses immediately.

BYOL and SimSiam: no negatives

Both avoid collapse without any repulsive term.

BYOL. An *online* network predicts the output of a *target* network. The target is an EMA of the online network. Only the online network receives gradients.

SimSiam. Shows the EMA is not essential. What matters is:

- an asymmetric **predictor** MLP on one branch, and
- a **stop-gradient** on the other

Remove either and the network collapses immediately.

Why this works is still not settled. The leading account frames it as an alternating optimisation resembling expectation-maximisation. Read SimSiam's ablations — the empirical finding is much stronger than the explanation.

DINO

Self-**di**stillation with **no** labels. Student and teacher are the same architecture; the teacher is an EMA of the student; the student matches the teacher's output distribution over augmented views.

DINO

Self-**di**stillation with **no** labels. Student and teacher are the same architecture; the teacher is an EMA of the student; the student matches the teacher's output distribution over augmented views.

Collapse is avoided by **centring** (subtract a running mean) and **sharpening** (low teacher temperature) — two opposing forces.

DINO

Self-**di**stillation with **no** labels. Student and teacher are the same architecture; the teacher is an EMA of the student; the student matches the teacher's output distribution over augmented views.

Collapse is avoided by **centring** (subtract a running mean) and **sharpening** (low teacher temperature) — two opposing forces.

The striking result: when the backbone is a ViT, the attention maps of the final layer segment objects — foreground from background, without ever having seen a mask.

DINO

Self-**di**stillation with **no** labels. Student and teacher are the same architecture; the teacher is an EMA of the student; the student matches the teacher's output distribution over augmented views.

Collapse is avoided by **centring** (subtract a running mean) and **sharpening** (low teacher temperature) — two opposing forces.

The striking result: when the backbone is a ViT, the attention maps of the final layer segment objects — foreground from background, without ever having seen a mask.

Supervised ViTs do not show this nearly as clearly. Something about the objective, not just the architecture, produces it.

Emergent segmentation in DINO

[Figure placeholder]

Final-layer attention maps from a DINO-trained ViT: object boundaries appear without the model ever seeing a segmentation mask

Masked image modelling

A different tradition, imported from language modelling: mask part of the input and reconstruct it.

Masked image modelling

A different tradition, imported from language modelling: mask part of the input and reconstruct it.

MAE makes it work for images:

- Mask a **very** high fraction of patches — 75%, against BERT's 15%
- The encoder sees **only the visible patches** — so it is cheap
- A lightweight decoder reconstructs pixels from encoded patches plus mask tokens
- The decoder is discarded after pretraining

Masked image modelling

A different tradition, imported from language modelling: mask part of the input and reconstruct it.

MAE makes it work for images:

- Mask a **very** high fraction of patches — 75%, against BERT's 15%
- The encoder sees **only the visible patches** — so it is cheap
- A lightweight decoder reconstructs pixels from encoded patches plus mask tokens
- The decoder is discarded after pretraining

Why so high a ratio? Images are far more redundant than text. Mask 15% and interpolation solves it. The task must be hard enough to require semantics.

Contrastive versus masked

They encode different priors, and it shows.

Contrastive versus masked

They encode different priors, and it shows.

- **Contrastive** methods learn invariance directly, so features are already linearly separable — strong linear probes, weaker fine-tuning gains.
- **Masked** methods learn to model image structure, so features are richer but less linearly organised — weaker probes, stronger fine-tuning.

Contrastive versus masked

They encode different priors, and it shows.

- **Contrastive** methods learn invariance directly, so features are already linearly separable — strong linear probes, weaker fine-tuning gains.
- **Masked** methods learn to model image structure, so features are richer but less linearly organised — weaker probes, stronger fine-tuning.

- Contrastive needs carefully designed augmentation; MAE needs almost none.
- MAE scales more gracefully with model size.

Contrastive versus masked

They encode different priors, and it shows.

- **Contrastive** methods learn invariance directly, so features are already linearly separable — strong linear probes, weaker fine-tuning gains.
- **Masked** methods learn to model image structure, so features are richer but less linearly organised — weaker probes, stronger fine-tuning.

- Contrastive needs carefully designed augmentation; MAE needs almost none.
- MAE scales more gracefully with model size.

Neither dominates. Which you want depends on whether you will fine-tune, and on how much you trust your augmentations.

Looking at representations

Before trusting a representation, inspect it.

- **t-SNE / UMAP** — project to 2D and look for class structure. See the t-SNE notes.
- **k-NN retrieval** — what are an image's nearest neighbours? Failures here are informative.
- **Linear probe per layer** — where does the useful information live?

Looking at representations

Before trusting a representation, inspect it.

- **t-SNE / UMAP** — project to 2D and look for class structure. See the t-SNE notes.
- **k-NN retrieval** — what are an image's nearest neighbours? Failures here are informative.
- **Linear probe per layer** — where does the useful information live?

A caution on t-SNE: cluster *sizes* and inter-cluster *distances* are not meaningful, and structure appears at some perplexities and not others. It is a tool for generating hypotheses, not for confirming them.

What self-supervision does not fix

Removing labels does not remove bias — it removes the *audit point*.

What self-supervision does not fix

Removing labels does not remove bias — it removes the *audit point*.

- The data still encodes whatever the collection process encoded: geographic skew, demographic skew, what the internet chooses to photograph.
- With no labels, there is no annotation schema to inspect for bias.
- Augmentation choices impose invariances that may be actively harmful for some groups or tasks — e.g. colour invariance and skin tone.
- Scale amplifies: these models become *foundations*, and their biases are inherited by everything built on them.

What self-supervision does not fix

Removing labels does not remove bias — it removes the *audit point*.

- The data still encodes whatever the collection process encoded: geographic skew, demographic skew, what the internet chooses to photograph.
- With no labels, there is no annotation schema to inspect for bias.
- Augmentation choices impose invariances that may be actively harmful for some groups or tasks — e.g. colour invariance and skin tone.
- Scale amplifies: these models become *foundations*, and their biases are inherited by everything built on them.

We return to this in Week 11.

Summary

- Labels are the bottleneck; self-supervision moves the prior from labels into augmentations and objectives
- Pretext tasks work partially, and are vulnerable to shortcut learning
- Contrastive learning (InfoNCE) needs negatives to prevent collapse
- SimCLR: augmentation, projection head, big batches. MoCo: momentum encoder and a queue instead
- BYOL and SimSiam avoid collapse with a predictor and stop-gradient
- DINO produces emergent object segmentation in ViT attention
- MAE reconstructs from 75%-masked inputs; strong under fine-tuning
- Evaluation protocol changes the ranking — read it first

For discussion next week

For discussion next week

1. SimCLR's projection head improves transfer by being *thrown away*. What does that tell you about what the contrastive loss is doing?

For discussion next week

1. SimCLR's projection head improves transfer by being *thrown away*. What does that tell you about what the contrastive loss is doing?

2. SimSiam shows stop-gradient prevents collapse but does not fully explain why. How much should that bother us? Is a reliable empirical finding without a mechanism good science?

For discussion next week

1. SimCLR's projection head improves transfer by being *thrown away*. What does that tell you about what the contrastive loss is doing?

2. SimSiam shows stop-gradient prevents collapse but does not fully explain why. How much should that bother us? Is a reliable empirical finding without a mechanism good science?

3. Contrastive learning requires you to name the nuisance variables in advance, through augmentation. Is that meaningfully less supervision than labelling?

For discussion next week

1. SimCLR's projection head improves transfer by being *thrown away*. What does that tell you about what the contrastive loss is doing?

2. SimSiam shows stop-gradient prevents collapse but does not fully explain why. How much should that bother us? Is a reliable empirical finding without a mechanism good science?

3. Contrastive learning requires you to name the nuisance variables in advance, through augmentation. Is that meaningfully less supervision than labelling?

4. You have 100,000 unlabelled chest X-rays and 500 labelled ones. Which method would you choose, and which augmentations would you refuse to use?

Copyright and License