Self-Supervised & Contrastive Learning

Lecture 17 · ES 667: Deep Learning

Prof. Nipun Batra
IIT Gandhinagar · Aug 2026

Learning outcomes

By the end of this lecture you will be able to:

1. State the **labeling bottleneck** and why self-supervision matters.
2. Describe **pretext tasks** and give 3 examples (rotation, jigsaw, colorization).
3. Write the **SimCLR pipeline** end-to-end · augmentations, projection head, InfoNCE.
4. Explain **InfoNCE** as a soft classification problem.
5. Contrast **SimCLR vs BYOL** and articulate why BYOL doesn't collapse.
6. Describe **MAE** (masked autoencoding) and when it beats contrastive.
7. Pick an SSL method for a given dataset scale and downstream task.

Where we are

Everything we've seen used labels — classification, MT, LLM pretraining on curated corpora. But labels are **expensive, finite, biased.**

Meanwhile, the internet has **unlimited unlabeled data**. Can we learn from it?

> REFERENCE
>
> Today maps to **UDL Ch 14** (unsupervised / contrastive). Papers: Chen 2020 (SimCLR), Grill 2020 (BYOL), He 2021 (MAE), Oquab 2023 (DINOv2).

Four questions:

1. What is self-supervised learning, formally?
2. How does **SimCLR** use augmentations as supervision?
3. Why does **BYOL** work **without negatives**?
4. How does **MAE** (masked autoencoding) compare to contrastive?

The labeling bottleneck · in numbers

DERIVATION

TASK	TYPICAL LABELING COST	TYPICAL DATASET SIZE
ImageNet class	0.5 USD per image (crowdsource)	14M images
Detection bbox	5 - 20 USD per image	200k images (COCO)
Segmentation mask	30 - 100 USD per image	10k images
Medical annotation	50 - 500 USD per image	~1 - 10k images

At 14M × $0.5, ImageNet cost ~$7M to label. Segmentation at that scale would be ~$500M. **Labels don't scale.**

Meanwhile · Common Crawl has 10^9+ web pages, Flickr has billions of photos, YouTube has zetabytes of video. All unlabeled.

PART 1

The labeling bottleneck

Why self-supervision scaled

Two facts about modern ML

1. **Labeled data is scarce.** ImageNet's 14M labels took thousands of human-hours. Medical imaging datasets struggle to reach 10k labeled cases.
2. **Unlabeled data is free.** YouTube uploads 500 hours / minute. The web has exabytes of text, images, video.

> KEY IDEA
>
> Self-supervised learning invents a "label" from the data itself — a surrogate task — and uses it to learn representations that transfer to real (labeled) downstream tasks.

Language already won with SSL — every LLM is pretrained with next-token prediction (which needs no labels). Vision caught up in 2020–2023.

What a surrogate task looks like

Given just an image (no label):

- **Predict the next word** (works for text, but images?)
- **Colorize a grayscale image** (labels are the original color channels)
- **Predict rotation** (0°/90°/180°/270°)
- **Fill in missing patches** (MAE)
- **Group augmented versions of the same image** (SimCLR, BYOL)

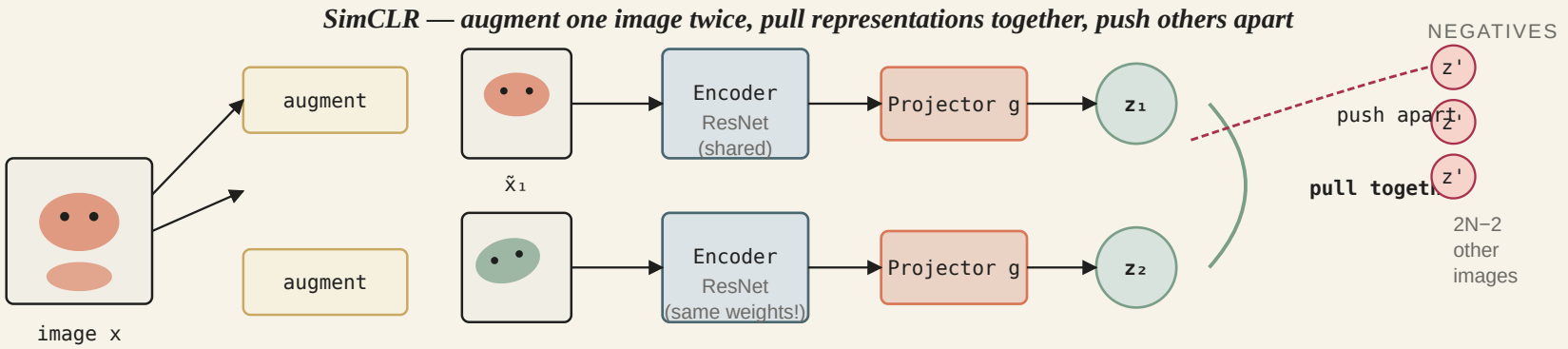
  All of these train an encoder to produce **useful representations** — features that later transfer to classification, detection, segmentation.

PART 2

Contrastive learning · SimCLR

Augmentations as implicit labels

The SimCLR framework



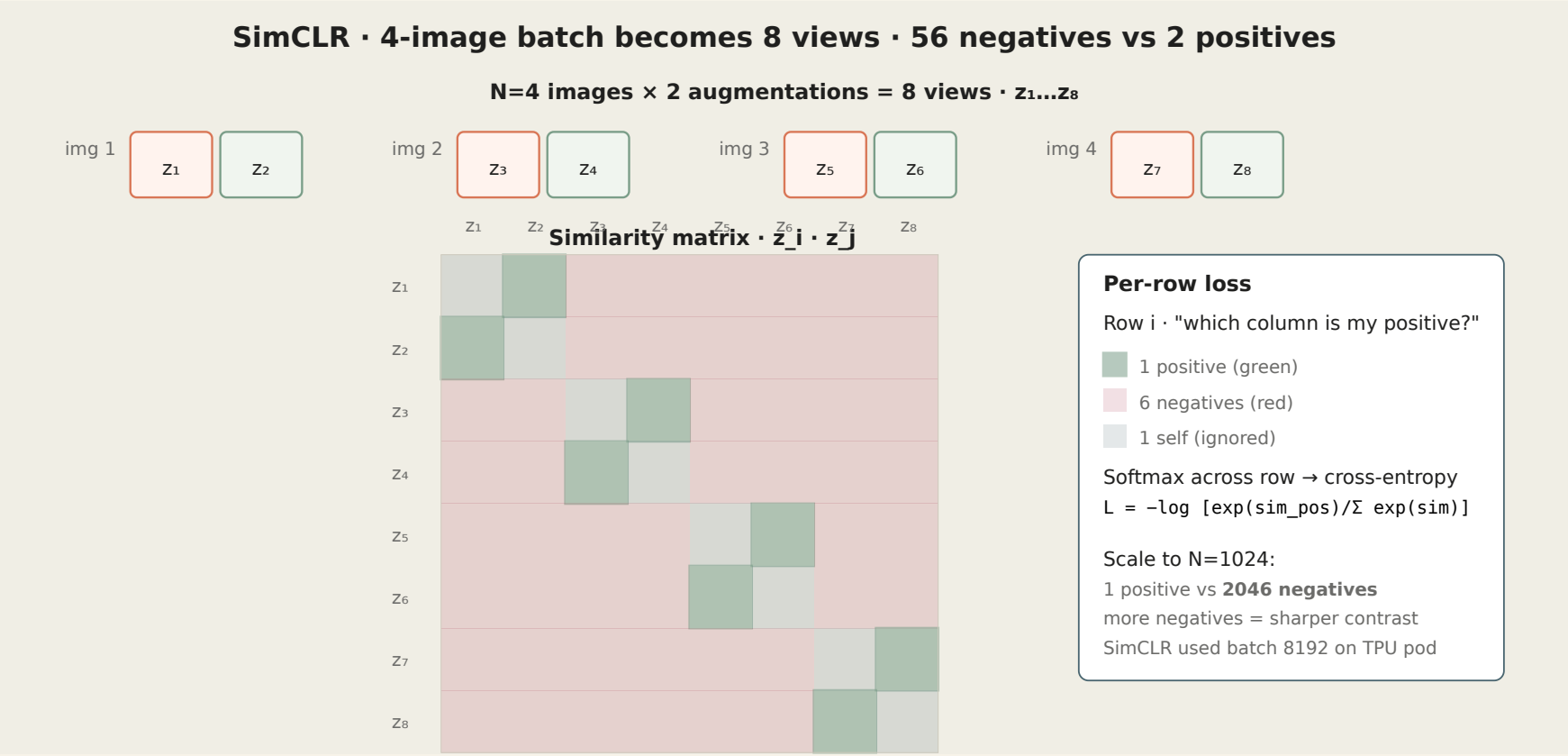
NT-XENT LOSS (NORMALIZED TEMPERATURE-SCALED CROSS-ENTROPY)

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j) / \tau)}{\sum_{k \neq i} \exp(\text{sim}(z_i, z_k) / \tau)}$$

numerator · the ONE positive pair · denominator · all 2N−1 other pairs

sim(a, b) = a·b / (‖a‖ ‖b‖) · τ ≈ 0.1 · larger batch → more negatives → better

SimCLR · batch as an 8×8 matrix



SimCLR · the matching-game analogy

KEY IDEA

You're given a huge pile of photos. Two of them are **your cat** (cropped differently, lit differently, color-jittered).

The task · find the two that match. Pick yours out of thousands.

That's contrastive learning. It pulls **same-image augmentations together** in feature space and pushes **everything else apart**. The model learns what makes "your cat" your cat — without anyone telling it the label "cat".

How SimCLR works, step-by-step

1. Sample a minibatch of N images.
2. Apply **two different augmentations** to each → $2N$ views.
3. The (i, j) pair from the same image = **positive**. All $2N - 2$ others = **negatives**.
4. Pass all through a **shared encoder** (ResNet / ViT).
5. Pass through a small **projection head** (2-layer MLP).
6. Compute similarity (cosine) in projection space.
7. Apply **NT-Xent loss** — pull positive together, push negatives apart.

InfoNCE · derive the loss step by step

For one anchor z_i with positive partner z_j in a batch with negatives z_k:

1. **Score function** · cosine similarity. Higher = more similar.
   $\text{sim}(z_i, z_k) = (z_i \cdot z_k)/(\|z_i\| \, \|z_k\|)$
2. **Make it a probability problem.** "Which z_k is the true partner of z_i?" Use softmax over scores, with **temperature** τ:
   $$P(\text{partner} = k) = \frac{\exp(\text{sim}(z_i, z_k)/\tau)}{\sum_{k' \neq i} \exp(\text{sim}(z_i, z_{k'})/\tau)}$$
3. **Cross-entropy** with the true partner j:
   $\mathcal{L}_{i,j} = -\log P(\text{partner} = j)$
4. **Substitute** to get the full form:

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k \neq i} \exp(\text{sim}(z_i, z_k)/\tau)}$$

It's standard softmax cross-entropy where the "classes" are batch positions and the label is the positive pair. No human labels needed.

Worked numeric · InfoNCE

Tiny batch of 2 images → 4 views. z_1, z_2 from image A; z_3, z_4 from image B. $\tau = 0.1$.

Compute loss for z_1 (positive · z_2, negatives · z_3, z_4).

Step 1 · similarities.

- $\text{sim}(z_1, z_2) = 0.9$ (positive)
- $\text{sim}(z_1, z_3) = 0.2$
- $\text{sim}(z_1, z_4) = -0.1$

Step 2 · scaled exps.

- $\exp(0.9/0.1) = \exp(9) \approx 8103.1$ (positive — also in denominator)
- $\exp(0.2/0.1) = \exp(2) \approx 7.4$
- $\exp(-0.1/0.1) = \exp(-1) \approx 0.4$

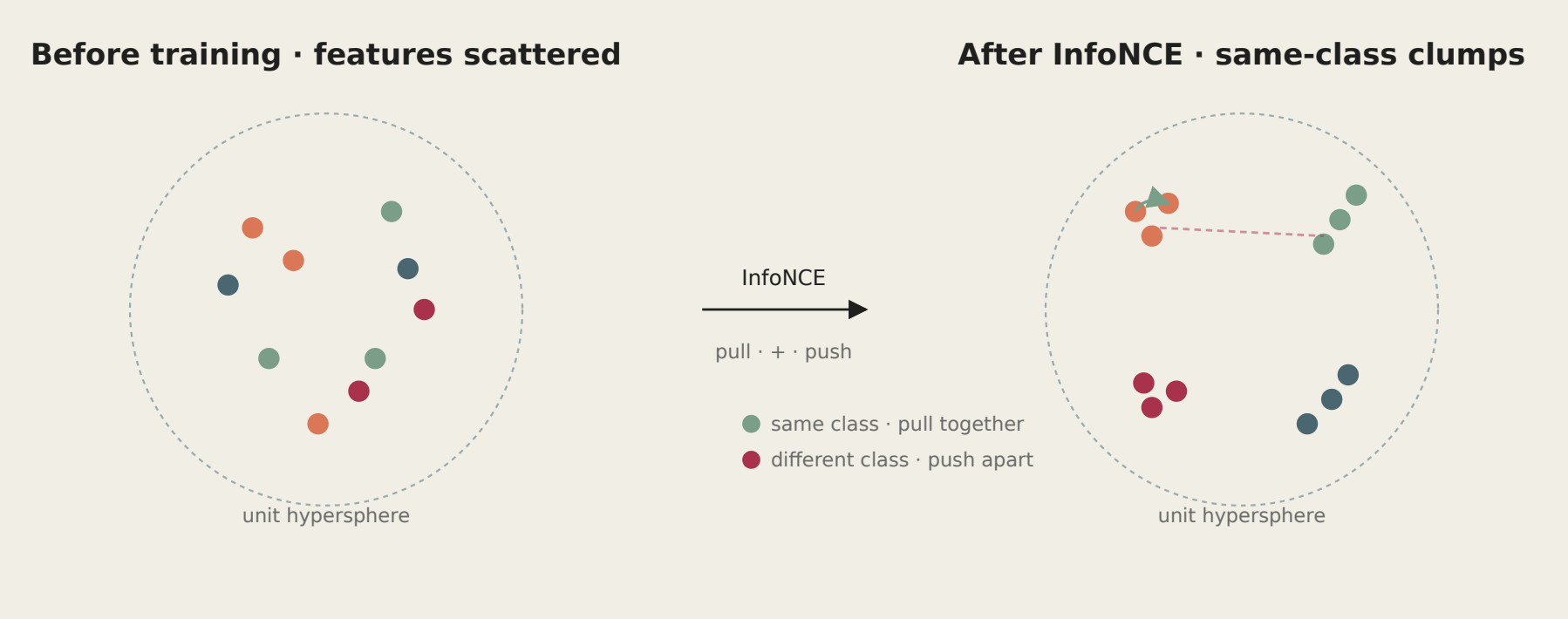
Denominator $\approx 8103.1 + 7.4 + 0.4 = 8110.9$.

Step 3 · loss.

$\mathcal{L} = -\log(8103.1/8110.9) = -\log(0.999) \approx \mathbf{0.001}$.

Loss is tiny because the positive's similarity dominates. If the model had assigned similarity 0.2 instead of 0.9 to z_2, the loss would be $\sim \log 3 \approx 1$ — gradient kicks in.

InfoNCE · geometrically



InfoNCE in plain English

Read the loss $\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_k \exp(\text{sim}(z_i, z_k)/\tau)}$ one piece at a time:

- **Numerator** — similarity to the *one* positive. We want this **high**.
- **Denominator** — sum of similarities to *all* $2N-1$ items in the batch. We want this **low** (except for the one positive already in the sum).
- **Temperature** τ — sharpens or softens the softmax. Smaller τ = sharper contrast, harder negatives matter more.

KEY IDEA

It's a **softmax classification** problem · "given z_i, which of the $2N-1$ candidates is its positive partner?" Cross-entropy with $2N-1$ classes, no labels needed — the positive is the *other augmentation of the same image*.

The crumple-zone projection head

InfoNCE forces the final reps to be **identical** for two augmentations of the same image. But what if a downstream task needs the info we just discarded (e.g. color matters for ripeness)?

INTUITION

Analogy · car crumple zone.

- **Encoder** f = passenger cabin. Should produce rich, general features h (color, texture, shape).
- **Projection head** g = crumple zone. Crushes h into z to satisfy the harsh contrastive task.
- **After pretraining** · throw away the crumple zone. Use h for downstream tasks.

This way the encoder doesn't have to delete useful info just to win the contrastive game.

Why SimCLR works

Three ingredients (Chen et al. 2020 ablations):

1. **Strong augmentations** — especially color jitter + random crop.
2. **Projection head** — throw it away after pretraining; the projection absorbs augmentation invariances.
3. **Large batch size** — more negatives → sharper contrast. SimCLR used batch 8192 on a TPU pod.

> IN PRACTICE
>
> Pretrained SimCLR features, fine-tuned, match or beat supervised ImageNet on many downstream tasks. Surprising in 2020; foundational today.

Temperature · the volume-knob analogy

INTUITION

Three critics give similarity scores $[0.9, 0.2, -0.1]$.

- **High τ (= 1.0)** · calm discussion. All voices heard.
- **Low τ (= 0.1)** · shouting match. The loudest voice (0.9) drowns out everyone.

In SSL, "hard negatives" are the most informative. Low τ makes the model **focus on them**.

Temperature · numeric demo

Same scores $[0.9, 0.2, -0.1]$, computed with two temperatures.

$\tau = 1.0$. Logits = scores.

- $\exp$: $[2.46, 1.22, 0.90]$. Sum $= 4.58$.
- Probs: $[0.54,\ 0.27,\ 0.19]$. Negatives still pull weight.

$\tau = 0.1$ **(SimCLR's choice).** Logits = scores × 10.

- $\exp([9, 2, -1])$: $[8103, 7.4, 0.4]$. Sum ≈ 8111.
- Probs: $[0.999,\ 0.001,\ 0.00005]$. The positive **completely dominates** → model is forced to make positive-similarity *much* higher than any negative.

Small τ ⇒ sharper distribution ⇒ hard negatives dominate the gradient. Used in SimCLR (~0.07) and as a learnable scalar in CLIP (L18).

Augmentation matters · *hugely*

Chen et al. swept augmentation pairs. Accuracy (linear-probe on ImageNet):

AUGMENTATION PAIR	ACCURACY
crop only	40%
color-jitter only	28%
crop + color-jitter	56%
crop + color + blur	64%

INTUITION

Contrastive learning is as much about **what invariances you pick** as about the loss. You're telling the model · "ignore crops, ignore color shifts, ignore blur — but pay attention to content." Those choices **become** the downstream invariances of the representation.

PART 3

BYOL · self-distillation without negatives

Two networks chase each other

BYOL · learning without negatives

SimCLR needed lots of negatives. What if we only **pull** positives together?

Danger · collapse. If the only force is "pull together," the model can output the same constant vector for every image · $f(x) = [0.5, 0.5, \ldots]$ → zero loss, useless features.

BYOL is a clever recipe to prevent collapse **without negatives**, using two asymmetric networks.

BYOL · twin networks (student + teacher)

- **Online network** ("student", θ) · trained with SGD. Has an extra **predictor** head.
- **Target network** ("teacher", ξ) · **not** updated by gradients. Weights are a slow EMA of θ.

Game · online sees view 1, predicts what target outputs for view 2.

Mechanism 1 · EMA target.
$\xi_{\text{new}} \leftarrow m\,\xi_{\text{old}} + (1-m)\,\theta, \quad m \approx 0.99$

Worked numeric · $m = 0.9$, $\theta_0 = [10, 2]$, init $\xi_0 = \theta_0$.

- Step 1 · $\theta_1 = [12, 3] \rightarrow \xi_1 = 0.9 \cdot [10, 2] + 0.1 \cdot [12, 3] = [10.2, 2.1]$
- Step 2 · $\theta_2 = [11, 5] \rightarrow \xi_2 = 0.9 \cdot [10.2, 2.1] + 0.1 \cdot [11, 5] = [10.28, 2.39]$

The teacher **trails** the student smoothly. The student is chasing a stable, slow-moving version of itself.

Mechanism 2 · stop-gradient on the target. Loss = `MSE(online_pred, sg(target))`. Gradients flow back through online only. The teacher can't "cheat" by moving its output to match the student.

The asymmetry (predictor + EMA + stop-grad) prevents collapse without needing negatives.

Why doesn't BYOL collapse?

Without negatives, the obvious failure mode is $f(x) = \text{const}$ — both networks output the same vector, loss is zero. Why doesn't this happen?

KEY IDEA

Three forces prevent collapse:

1. **Predictor head** introduces asymmetry — online must *predict* target, not match it directly.
2. **Stop-gradient** on the target prevents the target from moving to meet the online's output.
3. **EMA update** gives the target a momentum that trails the online; the online can never "catch" its target exactly.

Grill et al. had to run the training for 300 epochs to verify it really didn't collapse — nobody initially believed it.

MoCo, SwAV, and the contrastive zoo

The 2019–2021 era had many flavors:

METHOD	KEY IDEA
MoCo	FIFO queue of negatives; momentum encoder
SimCLR	large batch negatives; projection head
SwAV	online clustering (prototype assignments)
BYOL	no negatives; predictor + EMA
SimSiam	BYOL minus EMA — even simpler
Barlow Twins	decorrelate representations across views

By 2023 the community mostly converged on masked autoencoding (MAE) and self-distillation (DINO).

Linear probe vs fine-tune · the chef-and-knife analogy

INTUITION

We've forged a new chef's knife (the pretrained encoder). How do we test its quality?

- **Linear probe** · give the knife to a beginner and ask them to slice a tomato. Their technique is fixed and weak. If the cut is clean → the knife itself must be sharp. **Tests the inherent feature quality.**
- **Fine-tune** · give the knife to a master chef. They use all their expertise to get the best slice. **Tests max potential**, but a great chef can hide a mediocre knife.

METHOD	WHAT'S MEASURED	WHAT'S FROZEN
Linear probe	inherent feature quality	encoder frozen; only 1-layer classifier trained
Fine-tune	ceiling of the representation	nothing frozen (often low LR for encoder)
k-NN	local feature structure	encoder frozen; no classifier
Few-shot	sample efficiency	encoder frozen; tiny labeled set

Linear probe is the **cleanest** measure — it isolates the encoder. Fine-tune tests the ceiling but can hide a weak encoder.

2026 SSL benchmarks · who wins what

DERIVATION

BACKBONE	LINEAR PROBE IMAGENET	FINE-TUNE DETECTION
Supervised ResNet-50	76.1	38 mAP
SimCLR ResNet-50	69	36
MoCo-v3 ViT-B	76	39
MAE ViT-H	76	**54 mAP**
DINOv2 ViT-L	**86**	52

Observation · DINOv2 (self-distillation at scale, 142M images) dominates linear probe. MAE wins detection. Supervised is no longer SOTA for any frozen-feature evaluation.

The pattern across all these methods

Zoom out. Every SSL method is a variation on **"create a task the model can only solve if it learns features."**

KEY IDEA

- **Predict the future** (GPT, word2vec) — temporal / sequential structure.
- **Predict the missing part** (BERT, MAE) — contextual structure.
- **Match two views of the same thing** (SimCLR, MoCo) — invariance to nuisance.
- **Predict what a teacher thinks** (BYOL, DINO, knowledge distillation) — relational structure.

The architecture and loss differ, but the meta-idea is the same · **make the data supervise itself**.

PART 4

MAE · BERT for pixels

Masked autoencoding for images

MAE · the jigsaw-puzzle analogy

KEY IDEA

Take a photograph · **shred 75%** of it · ask someone to reconstruct the missing pieces.

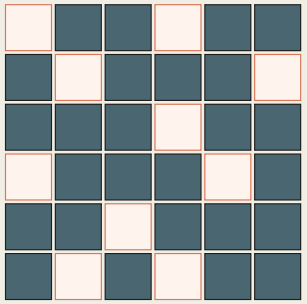
To do this they can't just look at local pixels · they must *understand* what a face looks like, what a tree branch is shaped like.

MAE forces the encoder to learn this **deep visual world model** by predicting the missing 75% from the visible 25%. It's BERT-for-pixels · masked-then-reconstruct as a self-supervision recipe.

MAE · full pipeline

MAE · mask 75% · heavy encoder on visible 25% · lightweight decoder reconstructs

1. Patchify + mask



75% masked · 25% visible
orange = visible · dark = masked

2. Heavy encoder · only visible tokens

ViT encoder

processes only 25% of patches

(huge speedup)

attention O((N/4)²) · 16× cheaper

3. Lightweight decoder · + mask tokens

small decoder

8 blocks vs encoder's 24

reconstructs masked patches

MSE loss on pixels

4. At inference · keep encoder only

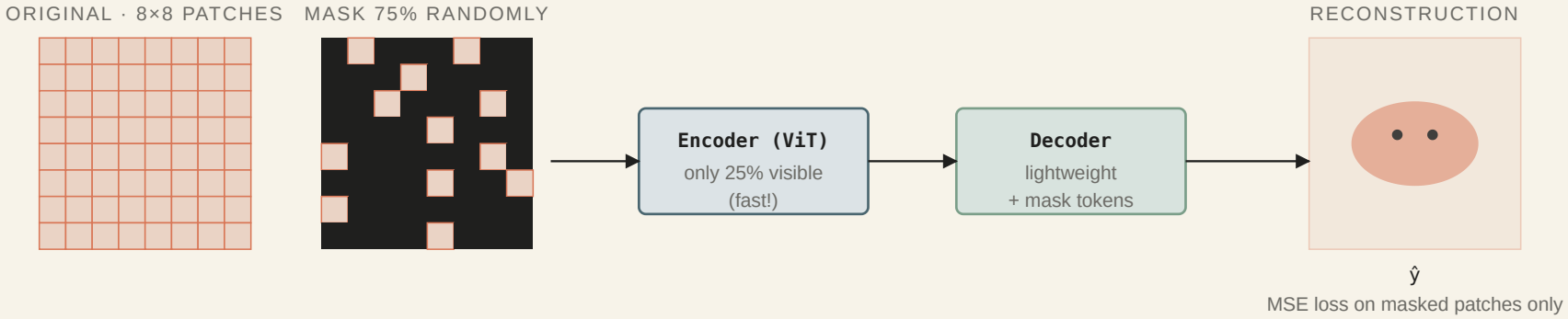
discard decoder · encoder features used for downstream tasks
linear-probe ImageNet · 76% (ViT-H · no finetune needed)

Why 75% works · images are highly redundant · 25% of patches is enough to reconstruct

He et al. 2021 · asymmetric encoder-decoder · SimCLR used 50% · MAE pushed it to 75% with better features

MAE · mask 75%, reconstruct the rest

Masked Autoencoder (MAE) — mask 75% of patches, reconstruct from 25%



KEY INSIGHT

Masking 75% (aggressive!) forces the encoder to learn **semantic** features, not pixel-level cues.

He et al. 2021 · ViT-Huge MAE pretraining matches supervised ImageNet on downstream tasks. Used in DINOv2, video understanding.

MAE · the asymmetric architecture

INTUITION

Analogy · the expert and the intern.

- Hire a world-class **expert** (heavy ViT encoder) to analyse a 100-page document.
- To save money, only show them **25 pages** (the visible patches).
- Hand the expert's brilliant summary to a cheap **intern** (lightweight decoder) and ask them to write a plausible version of the full 100 pages (reconstruct masked patches).

Why this is brilliant. The encoder — the expensive part — runs on **only 25% of the input** → ~4× faster pretraining than processing the full image. The decoder is small and only handles reconstruction.

MAE vs contrastive · who wins what

Contrastive (SimCLR)

- Needs massive batches for negatives
- Sensitive to augmentation choice
- Strong on classification
- Weak on dense tasks (segmentation)

MAE

- Any batch size works
- Minimal augmentation
- Strong on detection / segmentation
- Slightly weaker on pure classification

He et al. 2021 · ViT-Huge MAE pretraining → SOTA on many downstream vision tasks.

When should you use SSL?

DERIVATION

SCENARIO	USE SSL?
Plenty of labeled data, single task	No · just supervised
Large unlabeled pool, small labeled	**Yes · SSL pretrain + fine-tune**
Need generic features for many tasks	**Yes · start from DINOv2**
Need fast deployment on consumer GPU	Probably not · use CLIP/DINOv2 frozen
Novel domain (medical, satellite)	**Yes · in-domain SSL** then fine-tune

INTUITION

Rule of thumb (2026) · if labels cost more than compute, use SSL. In most real-world contexts, labels ARE the bottleneck. SSL tilts the equation.

SSL in text · the original success

GPT is self-supervised · next-token prediction on a trillion-token corpus. BERT is self-supervised · masked-LM.

KEY IDEA

All LLMs are self-supervised models, pretrained without a single human label (before RLHF tuning). The text modality has had SSL baked in since word2vec (2013). Vision caught up only around 2020 (SimCLR, MAE).

Contrast · NLP went straight to SSL because text is abundant and labels are expensive. Vision started supervised because ImageNet was cheap at 1M labels. The convergence · both modalities now use SSL as the foundation.

DINO and DINOv2 · self-distillation at scale

DINO (Caron et al. 2021) combined MAE-style patches with BYOL-style self-distillation for ViTs. Emergent properties:

- **Zero-shot object segmentation** appears in attention maps without any training on masks.
- Features transfer across domains without fine-tuning.

DINOv2 (Oquab et al. 2023) scaled this up to ViT-g (1B params) on 142M curated images.

> IN PRACTICE
>
> DINOv2 features are the *de facto* general-purpose vision representation in 2026 — ship it for any vision task where you can't afford full fine-tuning.

PART 5

Where self-supervision lives in 2026

The landscape

MODALITY	DOMINANT SSL APPROACH
Text	Next-token prediction (every LLM)
Vision	MAE + DINO-style distillation
Speech	Wav2Vec 2.0 / HuBERT (masked frame prediction)
Video	MAE extended to spacetime patches
Multimodal	CLIP-style contrastive image-text (next lecture)

INTUITION

Self-supervision created the **foundation model** era. Every modality now has its own canonical SSL recipe. Supervised learning survives only at the end of the pipeline — fine-tuning on small labeled data.

Summary · Lecture 17 — summary

- **Self-supervision** turns unlabeled data into training signal via surrogate tasks.
- **SimCLR** · pull two augmentations of the same image together, push all others apart. Needs large batches (for negatives).
- **BYOL** · two networks with EMA + stop-gradient; no negatives needed. Still works.
- **MAE** · mask 75% of patches; reconstruct; asymmetric encoder-decoder; the 2022 winner.
- **DINO(v2)** · self-distillation for ViTs; emergent segmentation in attention; the 2026 general-purpose vision representation.

Read before Lecture 18

Prince Ch 12 §12.5 (ViT) + CLIP paper (Radford 2021).

Next lecture

Vision-Language Models — ViT, CLIP, LLaVA, multimodal LLMs.

NOTEBOOK

Notebook 17 · `17-simclr-mini.ipynb` — implement NT-Xent from scratch; pretrain on CIFAR-10; t-SNE the embeddings to see class clustering without labels.