Self-Supervised Learning

The line between **unsupervised** and **self-supervised** learning *is* blurry, and people often use them interchangeably, even in papers 😄. But there **is** a subtle and important difference.

🔍 The Key Distinction

Aspect	Unsupervised Learning	Self-Supervised Learning
Labels	No labels at all	Still no human labels, but generates *pseudo-labels* from data
Goal	Discover structure in data	Learn representations useful for downstream tasks
Examples	Clustering, PCA, autoencoders	Contrastive learning, masked autoencoders, DINO, SimCLR
Tasks	Find patterns, groups, density	Solve a *pretext task* designed by us (e.g., match image views)

🧠 Intuition

- **Unsupervised** = "Let the model figure things out on its own"
  - No supervision of any kind
  - Model learns latent structure or clusters
- **Self-supervised** = "We *create* a task for the model to learn from itself"
  - No human labels, but we invent artificial tasks
  - E.g., "Predict the missing patch," "Match two augmented views"

---

semi-supervised learning sits right between **supervised** and **unsupervised** learning. It's super practical and widely used when **labeled data is scarce**, but we have **tons of unlabeled data** (which is often the case in real-world applications).

🧠 What is Semi-Supervised Learning?

Semi-supervised learning = using a small amount of labeled data + a large amount of unlabeled data to improve model performance.

- You start with some ground-truth labels (e.g., 10% of your data)
- Then you **leverage unlabeled data** to learn better representations or predictions
- This allows the model to **generalize better** without needing to label everything manually

---

⚖️ It's a Mix Of:

Component	Description
Supervised	Learns from labeled data
Unsupervised	Finds structure in unlabeled data
Semi-supervised	Uses both to get the best of both worlds

📚 Real-World Examples

Here are some **semi-supervised learning techniques and real examples**:

---

1. Pseudo-Labeling (a classic)

- Train a model on labeled data
- Use it to predict labels for unlabeled data
- Add high-confidence predictions as *pseudo-labels*
- Retrain with both real + pseudo-labeled data

📍Used in: classification, object detection, e.g. **YOLOv5 semi-supervised**

In **DINO**, all the augmented views (patches/crops) *do* come from the **same image** — so why doesn't the model just collapse to predicting the **same thing for everything**?

Let's unpack it step-by-step. This is one of the coolest ideas in modern self-supervised learning 🔥

⚠️ The Collapse Problem

In self-supervised learning, **collapse** means:

> The model outputs the **same representation for every input**, regardless of what it sees.

This can happen **easily** when the model is trained to match outputs of augmented views — like in DINO, SimCLR, BYOL — because there are **no negative pairs** to push things apart.

So the natural worry is:
👉 *"If we only tell the model to make all views of an image look the same... won't it just output a constant vector?"*

💡 Why DINO Doesn't Collapse (Despite Only Using Positive Pairs)

DINO prevents collapse through a few smart design choices:

---

1. 🧑‍🏫 Momentum Teacher

- DINO has **two networks**: a **student** and a **teacher**
- The **teacher is not trained by backprop**. Instead, it's an **EMA (Exponential Moving Average)** of the student.
- So the teacher evolves slowly → provides a **moving target** that is stable but improves over time.

✅ Prevents the student from immediately matching itself (no shortcut)

---

2. 🧊 Sharpening the Teacher Output

- The teacher's softmax output is **sharpened** (via low temperature $T \ll 1T \ll 1$)
- This forces it to give **confident, peaked distributions** over representations

✅ Helps the student learn **discriminative** (not uniform) representations
✅ Prevents the student from collapsing to uniform outputs

---

3. 🪞 Multi-Crop Strategy

- The same image is augmented into **2 global crops** + **several local crops**
- Global crops are used by both student and teacher
- Local crops are only used by the student
- The student must **match its view to the teacher's view**, even across scales and perspectives

✅ Teaches the student to build **consistent and flexible representations**

---

4. 📊 Centering the Teacher Output

- DINO **centers** the teacher's output across the batch
- This makes sure that **no dimension dominates**
- Helps keep the feature distribution **balanced and stable**

✅ Prevents collapse to a single vector or dimension

💥 Result

Even though DINO only sees **positive pairs**, it avoids collapse and learns:

- Clustered, semantically meaningful features
- Linearly separable embeddings (good for k-NN, image retrieval, etc.)
- No need for labels or contrastive negatives at all

🧠 Summary Table

Collapse Prevention Trick	What It Does
Momentum Teacher (EMA)	Stabilizes training, breaks feedback loop
Sharpened Teacher Output	Prevents uniformity, encourages confidence
Centering	Ensures diversity across dimensions
Multi-crop Augmentations	Forces representation consistency across scales/views