Self-Supervised Learning

2026-05-06 · cheerful mango Haubentaucher

There is free lunch.

The Why

In the previous chapter we learned to reuse a pretrained backbone across tasks, and that foundation models extend reuse to tasks they were never explicitly trained on. This chapter generalises that idea, learning representations from raw data through a family of objectives that scale to the corpora behind today's foundation models.

Corpus. A dataset is curated and usually labelled, assembled with a specific task in mind. A corpus is the body of raw data itself, often scraped or harvested at web scale, with no task attached. Self-supervised learning is what lets a corpus take the place of a dataset.

Labels $\tilde{y}$ are the bottleneck. The internet, sensor networks, and scientific instruments produce orders of magnitude more raw data than any annotation budget can keep up with. Annotation is expensive, biased by the curator, slow to scale with model size, and tied to the specific domain that worked them out.

ImageNet absorbed years of human effort to assemble fourteen million labelled images. The open web holds a hundred billion images that no curator will ever touch. The asymmetry between labelled and unlabelled data is the structural fact that motivates this chapter.

Self-supervised learning extracts supervision from the data itself. Instead of asking a human what an image contains, we design a pretext task whose label can be read off directly from the data structure.

Chapter 6 showed one route to such a backbone without labels: The variational autoencoder distilled a representation from a reconstruction objective alone.

- Pretext tasks. The pretext task is scaffolding; the representations the network builds while solving it are the real product, and they transfer to downstream tasks the network was never trained for. The first generation of self-supervised methods built auxiliary objectives by hand: Rotate the image and predict the rotation, scramble the patches and reassemble them, fill in the colour of a grayscale photo.

- Contrastive learning. A second generation drops the hand-designed task and asks the network to recognise the same image

under different views. Two random crops of one photo should land near each other in the embedding space, while crops from different photos should land apart.

MASKED AND NEXT-TOKEN PREDICTION. The dominant pretext task at scale is to mask part of the input and ask the model to reconstruct or denoise it. This can be masked tokens in BERT [1], masked patches in MAE, or the next token in autoregressive language models.

[1]

SPAN MASKING. Single-token masking is often trivial, since twenty visible neighbours give away the missing token. Span masking removes contiguous chunks instead, forcing the model to reconstruct multi-token structure. SpanBERT and T5 made this the default in modern encoder-decoder pretraining.

EXAMPLE. Single-token: `She slipped on a yellow [?] peel and fell over`. Span: `She slipped on a [? ? ?] and fell over`, where the contiguous chunk is the three adjacent tokens `yellow banana peel` the model has to recover together.

CAUSAL NEXT-TOKEN PREDICTION. The decoder-only language model takes the same denoising principle and applies it strictly left to right. At every position, the model sees only the prefix and predicts the next token. GPT scales this objective to internet-sized corpora, and the resulting model is a foundation model whose representations transfer to almost every language task by prompting alone.

IN THE BEGINNING WAS THE WORD. Prompt injection, reasoning, and agentic tool use.

PREDICTING IN LATENT SPACE. Reconstructing pixels, or tokens, wastes capacity on texture and lighting the downstream task does not care about. Joint-embedding predictive architectures (JEPA) move the target into feature space instead. The teacher and the student share the same encoder, the teacher's weights are an exponential moving average of the student's to stabilize targets over time, and no labels need to enter the pipeline at any point. On every step the teacher runs a forward pass on the unmasked input and produces target representations of the masked region. The student sees a masked view of the same input sample, predicts those target representations in the latent space, and the loss is the distance between predicted and target features. There is no decoder and pure prediction happens in the encoder's feature space, so the method is efficient and scales well to large models and large corpora.

CHOOSE BY REGIME. Architecture tracks the downstream task: Encoder-only for understanding, decoder-only for generation, encoder-decoder such as T5 and BART for both at doubled cost.

IN ORDER TO MOVE FROM EXPENSIVE MANUAL ANNOTATION TO LEARNING FROM RAW DATA we have good reason to find ways to,

DINO predates and inspires the JEPA family. A momentum teacher produces soft targets in feature space, the student matches them with cross entropy, and no negatives or labels are needed. Self-supervised attention maps emerge that segment objects without supervision.

- design auxiliary tasks whose supervision signal is inherent in the data itself, so that scale is bounded by storage rather than by human effort.
- learn representations that capture semantic structure without any

human labels, transferable across downstream tasks the model was never explicitly trained for.

- recognise denoising as the unifying objective behind masked-token, masked-patch, and next-token prediction.
- match the masking pattern to the downstream regime, choosing between encoder-only, decoder-only, and encoder-decoder recipes when the task is understanding, generation, or both.
- predict in feature space rather than pixel space when raw reconstruction would spend capacity on signal the downstream task does not care about.

THE FUNDAMENTAL QUESTION this chapter answers is whether the structure of unlabelled data carries enough signal to teach a model what to look at, without any human ever telling it what to value.

Self-Reflection and Recap

SELF-REFLECTION questions to guide your thinking:

- What goes wrong when a pretext task is too easy, and how can the network exploit shortcuts that solve the task without producing useful representations?
- Why does the contrastive recipe of treating different views of the same input as similar, and views of different inputs as dissimilar, give a stronger signal than handcrafted pretext tasks such as rotation prediction?
- Why is the choice of what counts as a different view the design decision that determines which features the contrastive representation will treat as invariant?
- Why is denoising a useful frame for thinking about masked-token, masked-patch, and next-token prediction together rather than as separate methods?
- Where does the supervision signal come from in masked prediction?
- How does the architectural choice between encoder-only, decoder-only, and encoder-decoder follow from whether the downstream task is understanding, generation, or both?
- Why does the encoder-decoder recipe end up paying for both halves of the bargain in architecture cost?
- Why does masking a contiguous span force the model to learn more than masking the same number of scattered single tokens?
- Why is moving the prediction target from pixel space to feature space a useful trade-off when raw reconstruction would spend capacity on signal the downstream task does not care about?
- What role does the slow-moving averaged teacher play in producing a stable target the student can predict against, and why is the same loss not workable when the teacher equals the student at every step?
- Why is the latent-space prediction recipe still considered self-supervised even though there is a teacher network feeding the student?
- Across all the methods in this chapter, what is the common idea that lets each of them produce useful representations from raw data with no human labels in the loop?

Recap of key concepts:

- Self-supervised learning derives its supervision from the structure of the data itself, with the pretext task as scaffolding and the learned representation as the product.
- Pretext tasks evolved from handcrafted recipes such as rotation prediction and patch scrambling toward principled relational objectives such as contrastive learning over augmented views.
- Denoising unifies the dominant masked-prediction family: Hide part of the input as masked tokens, masked patches, contiguous spans, or the next token in a sequence, and train the model to reconstruct what was hidden.
- The architectural choice between encoder-only, decoder-only, and encoder-decoder tracks the downstream regime, with encoder-only for understanding, decoder-only for generation, and encoder-decoder for both at the cost of doubled architecture.
- Predicting in latent space sidesteps the irrelevant signal that raw pixel reconstruction would spend capacity on, with an exponential moving average teacher providing a stable target the student can predict against in feature space.

Teaser. A self-supervised backbone gives you a head start, not a finished classifier; the downstream task still wants labels. What signals and cues can you think of to put in the place of supervision.

Self-supervised learning softened the annotation bottleneck from one side. Instead of asking humans to label more data, the chapter designed objectives and tasks that read supervision off the data structure itself, and the resulting representations thus can be transferred to downstream tasks the model was never explicitly trained for. The bottleneck has another side, where supervision for dedicated downstream tasks still arrives, from humans, from auxiliary sensors coupled to the input, or from larger models standing in as labellers, but in the form of rough rules, heuristics, proxy signals, or noisy stand-in labels rather than clean per-example annotations.

The next chapter turns to weak supervision, where a label model learns to combine the noisy and conflicting rule outputs into a single training signal. Self-supervised learning trusted the structure of the data; weak supervision trusts the structure of the rules; both share the conviction that some signal is a signal.

feedback