# *Weak Supervision*

*2026-05-06 · cheerful mango Haubentaucher*

A WEAK SIGNAL, IS STILL A SIGNAL.

## *The Why*

IN THE PREVIOUS CHAPTERS we have seen how to represent generalizable structure off the data, and defaulted to small clean head bolted on the embedding. This chapter turns to the head itself, asking what to do in the absence of clean labels.

LABELS ARE RANDOM VARIABLES $\tilde{y}$. Not ground truth facts, just noisy realisations of an underlying truth $y$ that we never observe directly. The annotation pipeline is a measurement device, and like every measurement device it has a bias and a variance.

WEAK LABELS ARE DRAWN FROM HETEROGENEOUS SOURCES, each with its own coverage and reliability profile. Crowd annotations exhibit substantial inter-annotator disagreement. Heuristic rules attain high precision on cases anticipated by their author and provide low coverage elsewhere. Large Pretrained classifier models afford broad coverage but are systematically overconfident on inputs lying outside the training distribution. Physical sensor measurements and behavioural proxies serve as auxiliary signals, correlating with downstream events of interest without themselves constituting labels, yet both can be folded into the pipeline as labelling-function outputs. The composition of these sources varies by modality, with text, image, sequence and tabular settings each exhibiting characteristic patterns. A unifying abstraction subsumes all of them: A labelling function is a programmatic mapping from input to a label. Under this representation, disparate sources reduce to a single object class amenable to downstream or bidirectional aggregation.

HEURISTICS AS BENCHMARKS. Cheap to author, interpretable in their failures, and immune to training-distribution shift, heuristic baselines establish the performance floor a learned method must clear. The gap between a heuristic and a learned model quantifies whether the additional complexity is justified.

SENSORS AND PROXIES. Sensors such as vibration, touch, torque, current draw, lidar and you name it serve as proxies for machine failure, anomaly onset, or obstacle presence. Behavioural signals such as clicks, dwell times, completion rates, and purchase records serve as proxies for relevance, engagement, or user preference.

A JUDGE IS A LABELLING FUNCTION THAT EXPLOITS THE VERIFIER-GENERATOR ASYMMETRY. Discriminating a good answer from a poor one is structurally easier than producing a good one from scratch, the same gap that powers GAN discriminators and RLHF reward models. A language model judge consumes a candidate output and

DIE ODE. Try writing a four-line ode to a banana in the manner of Schiller. Then judge whether this one is any good:

Sei mir gegrüßt, du krumme Frucht,
in Gold und Schweigen eingehüllt;
es trägt die Reife stille Wucht,
bis süß dein Schicksal sich erfüllt.

returns a scalar or categorical verdict [1], sitting alongside symbolic and statistical labelling functions under the same uniform interface. Judges offer lower cost than human evaluation and broader coverage than regular-expression rules, while inheriting the calibration and stylistic biases of the underlying model. Process reward models generalise the construction to intermediate reasoning steps, scoring each step of a chain rather than only the final answer and so spreading supervision along the whole trajectory instead of concentrating it at the end.

[1]

AUXILIARY TASKS construct intermediate and spreaded supervision. If the main task is hard to label, find an easier auxiliary task that shares structure with it and train on that instead.

GENERATORS ARE LARGE MODELS THAT PRODUCE LABELLED EXAMPLES DIRECTLY. Where a judge discriminates, a generator synthesises both input and label. The canonical case is knowledge distillation: A teacher's softmax supplies richer supervision than a one-hot target, and a smaller student is trained to match it. The same recipe underlies synthetic-data and instruction-tuning pipelines at scale.

SNORKEL. Three moving parts: Engineers write labelling functions in Python that each vote on a data point or abstain, a label model learns the accuracies and correlations of those functions to turn their votes into a probability distribution over the true label, and a discriminative end model is trained on those probabilistic labels and generalises beyond the rules.

SOFT LABELS ARE THE NATURAL OUTPUT OF WEAK SUPERVISION. A soft label assigns a probability to each candidate class, preserving inter-class structure that an argmax would discard. In Snorkel-style pipelines this distribution is the posterior of the label model, which sees only the labelling-function outputs and never the inputs themselves. The end model, a high-capacity discriminator, trains on those soft posteriors with full access to the inputs but never sees the labelling functions, and so generalises to inputs on which no labelling function ever fired. The same soft-label construction recurs across judges, distillation, and label smoothing, where the source of the distribution differs but the training objective is identical.

ACTIVE SUPERVISION. The supervised complement to weak supervision: A small budget of clean labels is spent on the inputs from which the model stands to gain the most . Strictly supervised learning, but with the selection strategy as the lever. Acquisition criteria such as uncertainty sampling, query-by-committee, and expected model change rank candidates by anticipated information gain, and an annotator is queried only for the highest-ranked.

IN ORDER TO MOVE FROM THE FICTION OF GROUND TRUTH LABELS TO PRINCIPLED LEARNING UNDER NOISE we have good reason to find ways to,

- collect rules, crowd annotations, pretrained classifiers, and sensor proxies behind one labelling-function interface.
- exploit the verifier-generator gap with language-model judges as discriminators over candidate outputs.
- use larger models as generators, distilling a teacher's softmax into a smaller student.
- aggregate noisy votes through a label model and train an end model on the resulting soft posteriors.

THE FUNDAMENTAL QUESTION this chapter answers is how to compose imperfect sources into supervision a model can trust.

## *Self-Reflection and Recap*

SELF-REFLECTION questions to guide your thinking:

- Why are labels best treated as random variables $\tilde{y}$ rather than as ground truth, and what does that reframing buy you when you design a training pipeline?
- Why is it useful to treat heterogeneous label sources, from heuristic rules and crowd annotations to pretrained classifiers and sensor proxies, behind a single labelling-function interface, and what does the abstraction hide?
- When does a sensor measurement or a behavioural signal qualify as a labelling function, and when does the correlation it relies on fall apart?
- What is the verifier-generator gap, and how do GAN discriminators, RLHF reward models, and LLM-as-judge all sit on the same asymmetry?
- What calibration and stylistic biases does an LLM-as-judge inherit from its base model, and what does that mean for using it as a labelling function at scale?
- How does a process reward model spread supervision along a reasoning chain rather than concentrating it on the final answer, and why does that matter for training?
- Why does knowledge distillation transfer more information through a teacher's softmax than through one-hot labels, and where does the same recipe show up at scale?
- How does the label model in Snorkel recover each labelling function's accuracy and correlation structure without ever observing the true label?
- Why does the end model generalise to inputs on which no labelling function ever fired, and what would break this property?
- What do label smoothing, distillation, and label-model posteriors share, and where do they differ?
- Across rules, judges, distilled teachers, and label models, what is the common move that turns noisy partial signal into supervision a model can trust?

Recap of key concepts:

- Labels are random variables, not ground truth; any noisy source, heuristic rule, crowd worker, pretrained classifier, sensor measurement, or behavioural proxy, is unified behind one labelling-function interface.
- Judges exploit the verifier-generator gap: discriminating a good output from a poor one is structurally easier than producing a good one, and LLM judges, GAN discriminators, and RLHF reward models all sit on the same asymmetry. Process reward models extend the construction to intermediate steps, scoring each step of a reasoning chain so that supervision is spread along the whole trajectory rather than concentrated on the final answer.
- Generators turn larger models into labellers; knowledge distillation transfers a teacher's softmax to a smaller student, and the same recipe powers synthetic-data and instruction-tuning pipelines.
- The label model aggregates labelling-function votes into a probabilistic posterior over the true label, sees only the votes and never the inputs, and a high-capacity end model trains on those soft posteriors with full input access and so generalises beyond the rules.
- Soft labels recur across label-model posteriors, distillation, and label smoothing; the source of the distribution differs, the training objective is identical.

Weak supervision interfaced the annotation bottleneck from noisy partial supervision. Instead of deriving supervision from the data structure as self-supervised learning did, the chapter accepted noisy labels from rules, crowds, sensors, judges, and larger models, and combined it through a label model into soft posteriors that an end model could train against. Throughout, the auxiliary model was a labelling model: It answered what class an input belonged to and nothing else.

Teaser. How does the supervision problem change when model interacts with an environment and receives supervision not as per-example labels but as action-level signals such as demonstrations, feedback, corrections, or rewards rather than per-example labels?

The next chapter turns to interactive and reinforcement learning, where the model's predictions turn into actions which interact with an environment, which proactively influences the supervision cues it receives. The supervision is no longer a static label per input, but a dynamic signal that depends on the model's behaviour and the environment's response.

feedback