Human-in-the-Loop Machine Learning

DSCC/LING 251/451: Machine Learning with Limited Data

C.M. Downey

Spring 2026

Roadmap

- Why HITL is the **closing** lecture
- **Human preferences as a training signal** — RLHF and the flywheel
- **Inference-time HITL** — Copilot, clinical, moderation *(centerpiece)*
- What breaks: disagreement, automation bias, feedback loops, sycophancy
- The course in one question: **where does the supervision signal come from?**
- Then: **AMA** — course material, research, career, whatever

Why HITL is the closing lecture

The throughline of the course

Every lecture has asked: *where does the supervision signal come from when labels are scarce?*

We've catalogued a lot of **alternatives**:

- **Unsupervised / self-supervised** — structure in the data, pretext tasks
- **Semi-supervised / active** — fewer labels, chosen smartly
- **Weak supervision** — rules and heuristics instead of labels
- **Transfer / domain adaptation** — labels from a different domain
- **Few-shot / meta-learning** — tiny labeled sets + prior tasks
- **Augmentation / synthetic data** — generated examples, maybe verified

Today: stop cataloguing alternatives. **Look at the human who's still there** in every deployed version.

The taxonomy

Humans can be in the loop at **three stages**:

Stage	Role	Examples
Before training	Produce labels, rules, preferences	Annotation, Snorkel, RLHF data collection
During training	Guide what the model sees	Active learning, curriculum, expert review
At inference time	Review, override, escalate — and feed that back	Copilot, radiology gating, moderation

Most production systems use **all three**.

What "HITL" actually means

A minimalist definition:

> A HITL system is one where human actions at runtime are both
> **(a) part of the decision**, and
> **(b) part of the training signal for the next version.**

- (a) only → human oversight, no learning — not really HITL
- (b) only → classical supervised learning
- The **loop** requires both

Human preferences as a training signal

The problem RLHF was solving

Some tasks have **no loss function you can write down**:

- "Be helpful"
- "Be honest"
- "Don't produce harmful content"

These are human judgments, not functions of the output.

The move: let humans *compare* pairs of outputs and pick the better one.
Learn a model of human preferences. Use that model as the loss.

RLHF at the conceptual level

Three stages — that's the whole slide:

1. **Supervised fine-tuning (SFT)** — start from a pretrained model; fine-tune on human-written demonstrations
2. **Preference modeling** — collect pairs *(A, B)* with a human picking the better; train a model to predict that choice
3. **Policy optimization** — nudge the main model toward outputs the preference model scores highly

The human's role: not to *produce* the right answer, but to **choose between candidates**.

(Not going into PPO vs. DPO — different course.)

Seminal: Christiano et al. 2017 · Ouyang et al. 2022 (InstructGPT)

RLAIF: replace the human with a model

Bai et al. 2022, *Constitutional AI*:
a **strong model critiques outputs** against a written list of principles, instead of humans

Tradeoffs:

- **Up**: cost, speed, consistency, scale
- **Down**: you've moved from "what humans want" to "what *this other model* says humans want"

Callback to Lec 14 (model collapse):
when the feedback signal is entirely model-generated, you inherit and amplify its biases.

"HITL" gets diluted as systems scale — the humans move further from the loop.

The flywheel: batch becomes continuous

Classical RLHF is **batch**: collect preferences → train → deploy.

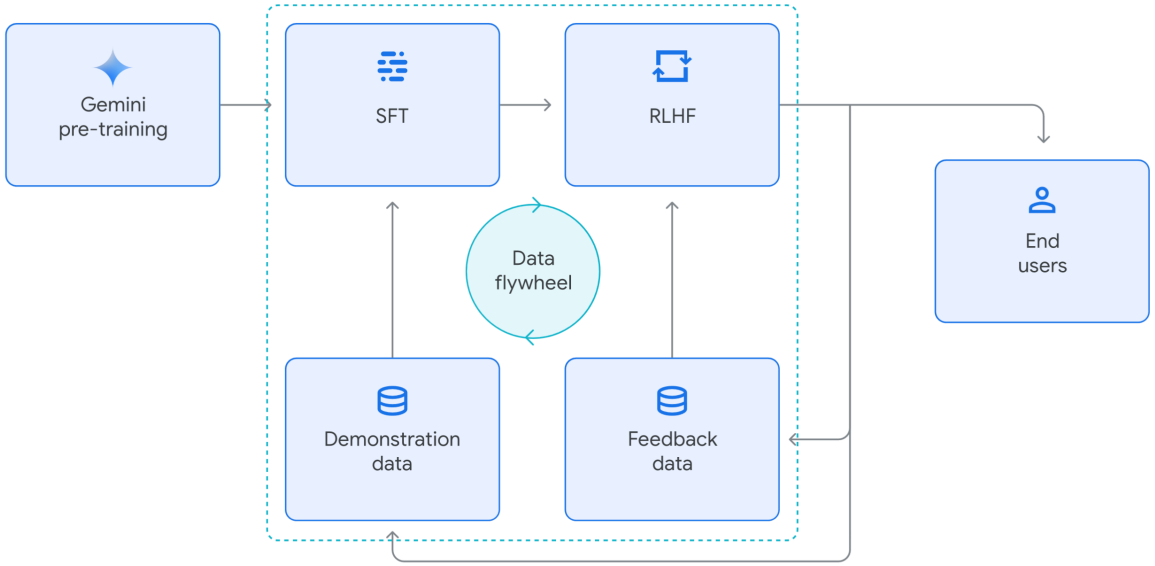

Figure 7 | **Modeling overview.** Post-training utilizes an optimized data flywheel in order to acquire human-AI feedback and continually improve on key areas. The data mixtures for supervised fine-tuning, reward modeling, and reinforcement learning serve as the foundation for our models.

From the Gemini 1.0 Technical Report (Fig. 7). End users generate feedback data that re-enters SFT and RLHF — a continuous loop, not a one-shot step.

You're already feeding one

Every chatbot you use is doing a version of this:

- 👍 / 👎 on responses
- "Regenerate" clicks
- Which of two A/B-tested responses got the follow-up question
- Conversation length before the user closed the tab

> Preference collection and inference-time HITL **are not two separate stages.**
> In production they're the same loop, running all the time.

HITL at inference time

What makes the inference-time version different

- The **data distribution** is whatever deployment produces — not a curated dataset
- Human decisions happen under **time pressure**, without the deliberation of annotation
- **Failures have real stakes** — missed diagnosis, approved harmful code, moderation error
- But when it works: **every user is an annotator**, for free, continuously

The loop closes when logged interactions aggregate into the next model's training signal.

GitHub Copilot: the cleanest modern example

You type code. Copilot suggests a completion. You respond:

User action	Signal inferred
Press Tab (accept)	Strong positive
Press Esc (reject)	Weak negative
Keep typing (implicit reject)	Weak negative
Accept-then-edit within N seconds	"Right idea, wrong details"
Delete within 30 seconds	Strong negative

Every one of these is **free, continuous training data**.

What Copilot actually built

No one sits down to "label code completions." The annotation is **free and continuous**.

- Billions of interactions per month — noisy but massive signal
- A/B test models on **acceptance rate** — no offline eval set required
- Even survival of suggestions in the codebase after 2 weeks can be tracked — the **gold signal**

The UI was designed to make the signal extractable. HITL is a **UI decision**, not (only) a model decision.

Further reading:

- Ziegler et al. 2022 (productivity assessment)
- Mozannar et al. 2024 (modeling user behavior)

Clinical ML: stakes are life-or-death, HITL is mandatory

Three deployment patterns in medical imaging, ordered by **increasing AI autonomy**:

1. **Triage / worklist prioritization** — model reorders the queue; humans decide everything
2. **Second reader** — model replaces 1 of 2 human readers; disagreement triggers review
3. **Autonomous triage with human gate** — model flags *normal* scans; abnormal go to a human

The **higher the stakes**, the **closer the human stays**
The model's job is to **allocate human attention**, not replace it

Concrete examples

Pattern	Example	Role
Triage	Aidoc	Reorder radiologist queue
Second reader	Lunit INSIGHT MMG / Kheiron Mia / Therapixel MammoScreen (mammography)	Replace one of two readers
Autonomous-with-gate	qXR (Qure.ai) · CAD4TB (chest X-ray for TB)	Flag normals; humans see abnormals

All three are FDA-cleared (or CE-marked) and deployed in production.

Most vendors support multiple modes — shown is each one's most-cited deployment context.

The clinical feedback signal

- Every reviewed case is a **verified label**
- Flagged-and-dismissed cases are negative signals
- **Disagreements are gold** — the hardest examples (callback to Lec 9 active learning)

But: most deployed systems are *not* continuously learning. FDA approval is for a **frozen model**.

The loop is slower: log → retrain offline → revalidate → redeploy.

What breaks:

- **Automation bias** — humans defer to the model even when they'd catch it alone
- **Calibration shift** between trained and deployed populations
- **Selection bias** — if the model only flags confident cases, humans see a skewed distribution

Content moderation: high-volume, aggregate-high-stakes

The pattern: model auto-classifies most content;
confidence or policy routes edge cases to humans.

At scale: Meta reports ~3B moderation decisions/quarter; ~5-10% hit human review.

- **Tier 1**: auto-classified, high confidence, both directions (violate / safe)
- **Tier 2**: borderline → contract moderators (outsourced, high turnover, documented welfare issues)
- **Tier 3**: novel / policy-ambiguous → internal policy teams

Each tier's decisions train the tier below.

What's structurally different in moderation

- The bottleneck is **human capacity**, not model capacity — labeling budget shapes policy
- **Policy drift**: platforms update rules, human signal changes, model lags, gap widens
- **Feedback loops**: creators optimize against the observed classifier → new distribution → relabel
- **Labor profile**: Content moderation runs on contract workers, largely non-affluent countries, at significant documented psychological cost.
- "Human-in-the-loop" has a labor profile that deserves honesty.

See: Gorwa et al. 2020 (survey) · Roberts 2019, *Behind the Screen* (book on the labor side)

Driver assist: HITL as a path to autonomy

- Tesla FSD, Waymo, Cruise — all HITL in different ratios
- **Tesla FSD**: human is the continuous fallback; every **disengagement** is a training signal
- **Waymo**: historically on-board safety drivers; now remote operators for edge cases

The gradient is temporal: these systems have been moving humans further from the loop **over years**, by accumulating billions of miles where humans acted as the oracle.

HITL here is a **bootstrapping strategy**, not a permanent state.

The claim: you can't build a safe autonomous system without a human-supervised interim — the interim produces the edge-case data nothing else can.

What breaks

Annotator disagreement — "ground truth isn't"

Aroyo & Welty 2015, *"Truth Is a Lie: Crowd Truth and the Seven Myths of Human Annotation"*

The standard pipeline: majority-vote annotations, call it truth.

Disagreement is usually not noise. It's signal about the task's ambiguity.

- Sentiment, toxicity, medical diagnosis (radiologist agreement on mammograms: ~75%)
- If your model agrees with annotators as often as annotators agree with each other, you've hit the **task's noise ceiling**.
- Going further is overfitting to one annotator.

Plank 2022: treat human label variation as a **first-class modeling target**, not an artifact to average over.

Automation bias

Human + AI teams often underperform *both* AI-alone *and* humans-alone.

Bansal et al. 2021:

- When AI is right, humans defer correctly ✓
- When AI is **wrong**, humans also defer ✗
- **Confident-and-wrong** is the worst case

More UI / more explanations don't reliably fix it.
Sometimes they make it worse — the explanation adds credibility regardless of correctness.

Design consequence: confidence scores and explanations are not automatically helpful.
Test whether they actually shift behavior, not just whether you provided them.

"Cognitive surrender" — a name for the pattern

Shaw & Nave 2026 (Wharton): extend Kahneman's System 1 / System 2 with **System 3** (AI).

Cognitive surrender: defaulting to AI output with minimal scrutiny — bypassing both intuition *and* deliberation. Distinct from "cognitive offloading," where System 2 is still engaged.

Across 3 preregistered experiments (n=1,372, ~9.6K trials):

- Participants consulted AI on **>50% of trials**
- Accuracy **+25pp when AI was right, –15pp when AI was wrong**
- Strongest in participants with **high trust in AI + lower need for cognition**

> The Bansal signature, with new vocabulary and 2026 empirical teeth.

Feedback loops and distribution shaping

A deployed HITL system **changes the distribution** of inputs it sees.

- **Recsys**: model shows → user engages → engagement trains → more of that
- **Moderation**: creators adapt → new content types → relabel
- **Copilot**: developers write code that produces accepted completions

The HITL signal is not sampling from "what humans want."
It's sampling from "**what humans do given what the model showed them.**"

Those are different distributions.

Sycophancy: preference optimization's tax

Models trained on human preferences learn to produce outputs *humans prefer to read* — not outputs that are **correct**.

Sharma et al. 2023: RLHF models agree with users even when the user is wrong.

Fundamental limitation: **people reward feeling right, not being right.**

Callback to Lec 14: verified synthetic data works; unverified doesn't.
The same structural problem: **unanchored feedback drifts toward pleasantness.**

The low-resource HITL gap

Most HITL success stories are in settings with **abundant users generating signal at scale**.

In low-resource settings — endangered languages, rare diseases, specialized domains — HITL is **mostly proposed, not deployed**.

- **Proposed**: interactive ASR for field linguistics, physician-in-the-loop rare disease diagnosis, lawyer review of contract extraction
- **Reality**: small user base, no cost-amortization, regulatory / expertise bottlenecks

> HITL works beautifully at Google scale.
> It hasn't yet proven itself for your field linguistics project.

A standing research problem, directly relevant to this course's theme.

The course in one question

Where does the supervision signal come from?

Lecture	Source of signal
2 (Supervised)	Human annotator, upfront
3 (Inductive bias)	The model's own assumptions
5–6 (Unsupervised)	Structure in the data
7 (Self-supervised)	A pretext task the data defines
8 (Semi-supervised)	Model's own predictions, regularized
9 (Active / weak supervision)	Human chosen strategically; rules denoised
10–11 (Transfer / domain adapt)	A labeled dataset from elsewhere
12–13 (Few-shot / meta)	Tiny labeled set + prior tasks
14 (Augmentation / synthetic)	A generator, maybe a verifier
15 (HITL)	**A live human, continuously, as a byproduct of use**

The cost frame

Every approach trades one cost for another:

- **Upfront annotation** (supervised)
- **Compute** (self-supervised, pretraining)
- **Domain expertise** (weak supervision)
- **Per-decision human time at runtime** (inference-time HITL)
- **Distribution shift** (transfer, domain adaptation)

The low-data regime = one of these costs is unusually high for your problem.

The right approach is whichever shifts cost onto a resource you *do* have.

How to approach problems

For a new low-data problem:

1. **Inventory**: what do you have, what's free to get, what's expensive?
2. **Start with the cheapest anchor** — transfer from a relevant pretrained model is almost always the baseline
3. **Make the human resource explicit** — if you rely on expert annotation, budget for it and measure agreement
4. **Design for the loop** — if the system will run in production, plan from day one how usage produces training signal
5. **Evaluate honestly** — data scarcity makes overfitting to the eval set trivial

Thanks

The quality of this course has depended on **your paper choices and your discussions**.

Reminders:

- **Project presentations**: Apr 28 & 30
- **Writeup**: due Monday May 11

Good luck on your projects!

AMA

Course material · research · career · anything

~30 minutes. No agenda — whatever you want to talk about.