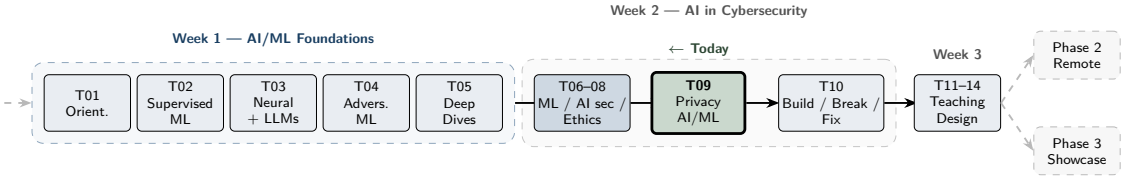
Privacy in AI/ML

AI + Cybersecurity Faculty Summer Institute

June 11, 2026

Where today sits in the institute

- Week 1 built the foundations: supervised ML, neural networks, LLMs, adversarial AI.
- Week 2 applied ML to security (T06), secured AI applications (T07), and worked through ethics and governance (T08); today we zero in on privacy.
- Tomorrow synthesizes via Build-it / Break-it / Fix-it (T10); Week 3 turns these into teaching modules; Phases 2 and 3 extend them into participants' own courses.



What is privacy?

Informational self-determination (Westin 1967): the right of individuals to control what others learn about them and what those others do with that information.

Questions for today:

- Where does private data show up in the ML pipeline?
- What can be learned about this data?
- How can we protect against this leakage?

Privacy vs. security. Security is about keeping unauthorized parties out. Privacy is about what happens to information *even when accessed legitimately*.

Prvacy Concerns in AI/ML

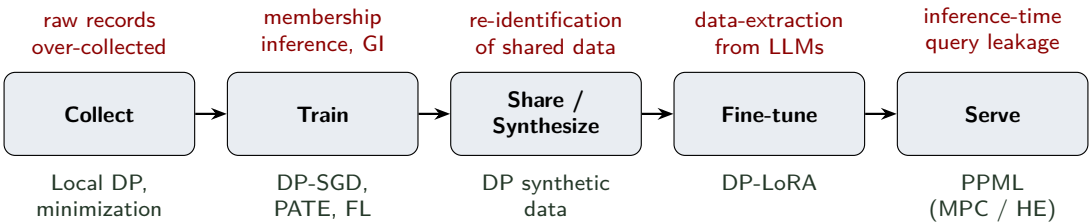
- ML systems are trained on **records about people** – medical histories, browsing logs, employee email, criminal-justice data, chat transcripts.
- Those records were collected under *some* expectation: a doctor, an employer, a platform with a stated policy. The training process changes who can see what.
- Once data enters a model, the usual privacy controls – access logs, role-based permissions, deletion – do not straightforwardly apply. The model becomes a new vehicle for whatever was in the data.
- This is now a deployment reality: hospitals training risk scores, banks training fraud detectors, vendors fine-tuning LLMs on customer support logs, agencies sharing models across jurisdictions.

Question for this morning: when a model is trained on sensitive data, what can leak, and what defenses actually constrain that leakage?

Where “naive” AI/ML falls short on privacy

- **Anonymization is not enough.** Stripping names and IDs does not prevent reidentification (Narayanan and Shmatikov 2008; De Montjoye et al. 2015).
- **A private dataset does not produce a private model.** Queries against the deployed model can reveal specific training records (Carlini et al. 2021; Shokri et al. 2017).
- **Access control stops at the model boundary.** Once weights are released or hosted, anyone with API access has access to the training data inside.
- **“We logged the prompt, not the training data” is also wrong.** Prompts and outputs are one leakage channel (Topic 07). The *model itself* is a separate one, and the defenses look different.
- **Utility vs privacy** If your model is too private, it won’t be useful. If it’s too useful, it won’t be private.

Privacy across the ML pipeline



Today's menu of defenses

Differential Privacy (DP)

formal guarantee on the *algorithm*

Federated Learning (FL)

structural defense: data stays on device

Secure Comp. for ML (PPML)

compute on encrypted data

Each defense answers a different version of "what is protected, from whom, and at what cost?"

Three running examples

Recidivism Risk Classifier

What it is. Tabular classifier trained on demographic + criminal-history records.

Privacy threat. Membership inference – "was this person in the training set?"

Defense. Differential Privacy.

Federated Phishing Filter

What it is. Multi-enterprise phishing classifier; federated across organizations that cannot share inboxes.

Privacy threat. Gradient inversion – a curious server reconstructs a client's training input.

Defense. Federated Learning.

Hosted Code Assistant

What it is. Code-completion LLM fine-tuned on private repos, hosted by an external vendor.

Privacy threat. Training-data extraction + queries crossing a trust boundary.

Defense. Secure Computation for ML.

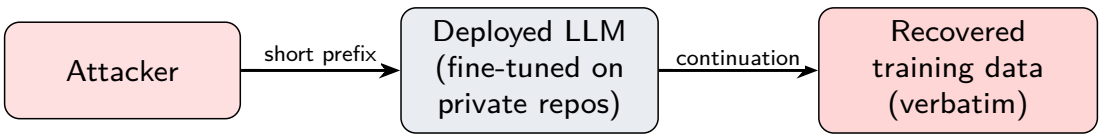
Outline

One Key Thing to Remember

- A model trained on private data is not, by itself, a private artifact.

Extracting training data from a deployed LLM

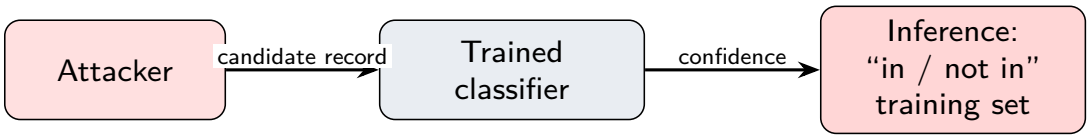
- **Setup.** Code-completion LLM fine-tuned on private repositories that include credentials and proprietary logic.
- **Attack.** Short prefixes elicit *verbatim* training-data continuations (Carlini et al. 2021; *First Principles* methodology, 2022).
- **Implication.** The *model* is the leak, not the prompt.



Black-box query access ⇒ verbatim recovery of training records (Carlini et al. 2021, 2022).

Membership inference

- **Setup.** Recidivism Risk Classifier: a tabular classifier trained on demographic + criminal-history records.
- **Attack.** Classical Shokri shadow-model MI; LiRA (Carlini 2022) as the modern methodology.
- **Impact.** An attacker learns *who was in the training set* – direct privacy harm to the people whose records were used.



Shokri et al. 2017; Carlini et al. 2022 *First Principles* formalizes the methodology.

This is not unique to LLMs

Training-data leakage takes several forms across model classes:

- **Tabular and image classifiers.** *Membership inference* – attacker probes the deployed model and infers whether a specific record was in training (Shokri et al. 2017; Hu et al. 2022 survey; Carlini et al. 2022 *First Principles*).
- **Federated training.** *Gradient inversion* – attacker on the server reconstructs a client's training input from the gradient update (Geiping et al. 2020; Huang et al. 2021 evaluation).
- **LLM fine-tuning.** *Training-data extraction* – response contains verbatim training data (Carlini et al. 2021; *First Principles* methodology, 2022).

Out of scope: model inversion and model stealing – both treat the model itself as the asset; these are secrecy / IP concerns, not individual privacy.

Time for a Demo

Outline

What was the problem?

- The model "memorized" the training data. – training-data extraction.
- Probability of correct classification differs for records in vs. out of the training set. – membership inference.
- The output of the model depends too much on any single training record. We need a way to constrain that dependence.

Differential Privacy (DP) limits the influence of any single training record on the model's output.

ε-DP definition

An algorithm M is ε-**differentially private** if, for all neighboring datasets D and D' (differing in one record) and all sets of outputs S:

$$\Pr[M(D) \in S] \;\leq\; e^{\varepsilon} \Pr[M(D') \in S]$$

- **Plain English.** Including or excluding any single record barely changes the probability of any output.
- ε small $\Rightarrow$ strong privacy; ε large $\Rightarrow$ weaker privacy.
- Dwork & Roth, *Algorithmic Foundations of DP* (2014) – the canonical reference.

Properties of ε-DP

An algorithm M is ε-**differentially private** if, for all neighboring datasets D and D' (differing in one record) and all sets of outputs S:

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \Pr[M(D') \in S]$$

- **Composability.** The total privacy budget is the sum of the budgets for each query.
- **Robustness.** DP provides a strong privacy guarantee that is independent of the adversary's prior knowledge.
- **Post-processing immunity.** Any function of a DP output is also DP (with no additional privacy loss).
- **Practicality.** DP mechanisms are often simple and easy to implement in real-world applications.

(ε, δ)-DP: a relaxed variant

An algorithm M is (ε, δ)-**DP** if:

$$\Pr[M(D) \in S] \;\leq\; e^{\varepsilon} \Pr[M(D') \in S] + \delta$$

- **Plain English.** Allow a small failure probability δ — the chance the privacy guarantee does not hold at all — to escape pure ε-DP's worst case.
- Typical deployed δ: cryptographically small (e.g., $\delta = 1/n^2$ for dataset size n).
- Why this variant matters for ML: the Gaussian mechanism (next slides) is (ε, δ)-DP but not pure ε-DP.

Composition: ε is a budget

Privacy budgets compose. The total ε spent across multiple queries is at most the sum:

$$\varepsilon_{\text{total}} \leq \sum_i \varepsilon_i$$

- **Plain English.** Each query spends some of the budget. They add up.
- **Implication.** Each query, each training epoch, each release spends some budget.
- **Example.** Each epoch of DP-SGD on the recidivism classifier spends ε_i; the privacy accountant tracks the total. Choosing $\varepsilon = 3$ means deciding how many training epochs you can afford.

ε	0.1	1	2	3	5	10
e^ε	1.11	2.72	7.39	20.1	148.4	22,026

Output probabilities can change by at most this factor when one record is added or removed.

ε in practice: deployed values

ε is a real number that real deployments choose.

Deployment	ε	**What it bought**
Apple emoji + word telemetry	$\approx$ 4–16 per day	On-device aggregation of usage patterns across hundreds of millions of devices
US Census 2020 disambiguation	≈ 19.61	Privacy of the entire decennial release; heavily debated

References: OpenDP deployed-ε registry; DifferentialPrivacy.org resource hub; Apple *Learning with Privacy at Scale* (2017); Kifer et al. 2023 *Epsilon** for "what does ε really mean."

Sensitivity Δf: how much one record can change the output

Sensitivity Δf is the maximum amount the function f's output can change if one record changes:

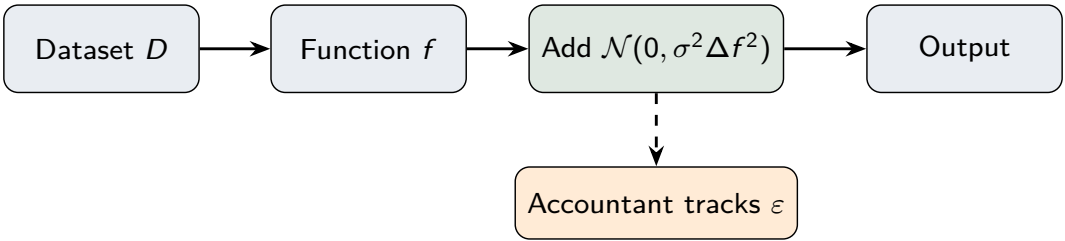
$$\Delta f \;=\; \max_{D, D' \text{ neighbors}} \|f(D) - f(D')\|$$

- **Intuition.** High sensitivity $\Rightarrow$ more noise needed for the same ε.
- **Example.** "How many records have property X?" $\Rightarrow \Delta f = 1$. One record can change the count by one.
- Sensitivity is the parameter that ties ε to the function being computed.
- **Example.** In DP-SGD, per-example gradient norms are clipped to $C \Rightarrow \Delta f \leq C$. Clipping bounds sensitivity.

The Gaussian mechanism

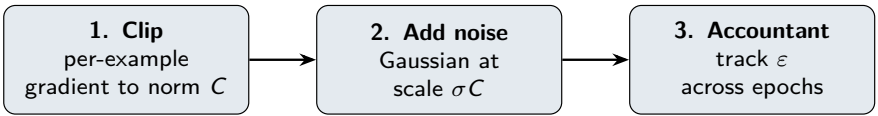
$$M(D) \;=\; f(D) + \mathcal{N}\big(0,\, \sigma^2 \Delta f^2\big)$$

- **Plain English.** Add Gaussian noise scaled by how much one record could change the answer.
- σ depends on ε and δ (standard relationship; see Dwork & Roth Appendix A).
- **Why Gaussian for ML?** Composes cleanly across many queries; works on continuous-valued gradients.



DP-SGD: the three ingredients

$$\tilde{g}_t \;=\; \frac{1}{L}\sum_{i\in B} \mathrm{clip}\big(g_t(x_i),\, C\big) + \mathcal{N}\big(0,\, \sigma^2 C^2 I\big)$$



Abadi et al. 2016. Clipping bounds sensitivity; noise enforces (ε, δ)-DP per step.

- **Clip:** per-example gradient norm bounded to C (this bounds Δf).
- **Noise:** Gaussian at scale σC – the mechanism.
- **Accountant:** tracks ε across epochs.

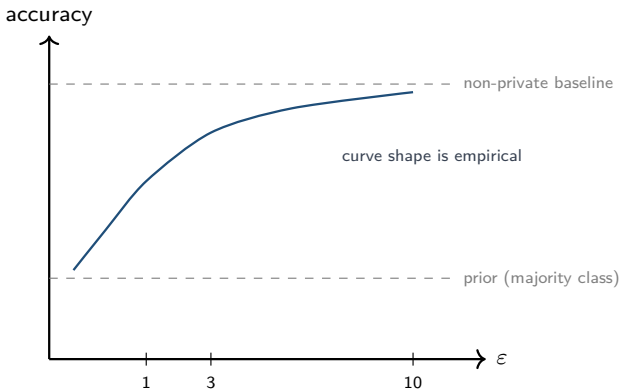
DP-SGD on The Recidivism Risk Classifier

Recidivism Risk Classifier under DP-SGD

- **Setup.** Tabular classifier trained with DP-SGD via Opacus at $\varepsilon = 3$.
- **Effect on attack.** MI success rate drops measurably (from $\sim$70% to near baseline 50–55%).
- **Effect on utility.** Classifier accuracy drops $\sim$2–5 percentage points (cf. Ziller et al. 2021 medical-imaging numbers as comparable benchmark).
- **What DP-SGD on A does not buy.** Defense against *gradient inversion* if the same model is trained in a federated setting – foreshadow the FL block.

This also applies to: DP-SGD on text classifiers (Example B); on LLM fine-tuning (Example C, research frontier per Cui et al. 2025).

The accuracy / ε tradeoff

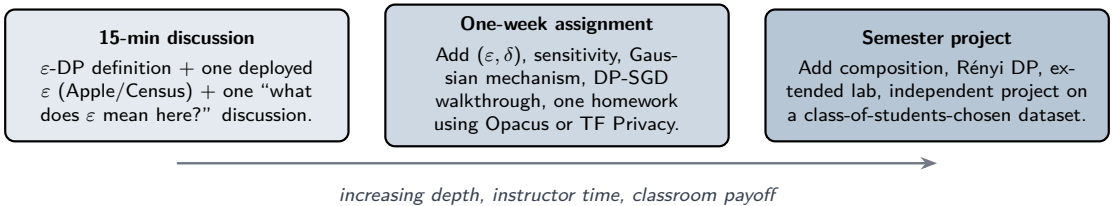


- Lower $\varepsilon \Rightarrow$ stronger privacy, weaker accuracy.
- At $\varepsilon = 1$: $\sim$5–15% drop on a typical tabular classifier.
- At $\varepsilon = 10$: $\sim$1–3% drop.
- Choosing ε is choosing where on this curve to live.

Numbers representative of tabular and imaging benchmarks; Ziller et al. 2021 (medical imaging); Anil et al. 2022 (BERT fine-tuning). Exact drop depends on architecture and dataset.

Back to the Demo

Teach-this-in-your-course: DP at three depth tiers

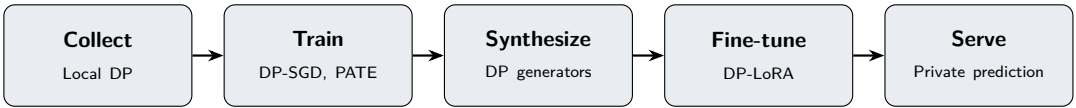


References to assign per tier: NIST DP blog (lecture); Cui et al. 2025 DP-survey (week); Dwork & Roth monograph (module). *DP-SGD also applies to text classifiers (B) and LLM fine-tuning (C, research-frontier).*

Outline

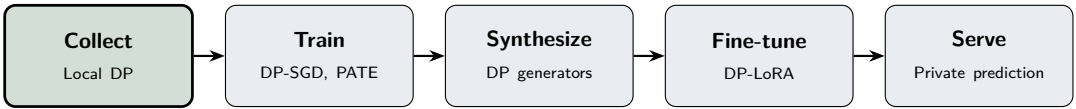
DP Across the ML Pipeline

Where you place the DP shield determines what gets protected and from whom.



Each stage admits a different DP technique. Today we walk through all five.

Collect



Collect (green): apply DP where data is generated, before it ever leaves the device.

Collect: Local DP at data collection

- **Mechanism.** Each device adds noise locally; the aggregator only sees noisy values. Aggregate statistics still recoverable at scale.
- **Deployed examples.** Google RAPPOR (Chrome telemetry); Apple iOS emoji + word telemetry.
- **Trust model.** No trusted aggregator required – strongest deployment story but highest noise cost.
- **Example.** An enterprise phishing classifier could use local DP on per-employee features before central training.

Local DP example: randomized response (Warner 1965)

The mechanism. To answer a sensitive yes/no question, each respondent:

- Flips a fair coin. **Heads** $\rightarrow$ answer truthfully.
- **Tails** $\rightarrow$ flip again. Heads $\rightarrow$ report "yes," tails $\rightarrow$ report "no."

The aggregator never learns any individual's truthful answer — each respondent has plausible deniability.

Recovering the population fraction. Let π = true "yes" rate, y = observed "yes" rate.

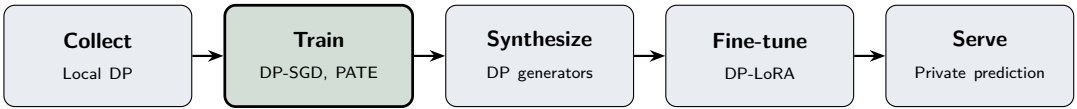
$$\mathbb{E}[y] \;=\; \tfrac{1}{2}\pi + \tfrac{1}{4} \quad\Longrightarrow\quad \hat{\pi} = 2y - \tfrac{1}{2}$$

With n respondents, the standard error of $\hat{\pi}$ is $O(1/\sqrt{n})$: noise washes out at scale, accuracy stays.

Privacy guarantee. Any report is at most $3\times$ as likely under one truthful answer than the other:

$$\frac{\Pr[\text{report "yes"} \mid \text{truth} = \text{yes}]}{\Pr[\text{report "yes"} \mid \text{truth} = \text{no}]} = \frac{3/4}{1/4} = e^{\varepsilon} \quad\Rightarrow\quad \varepsilon = \ln 3 \approx 1.10$$

Train



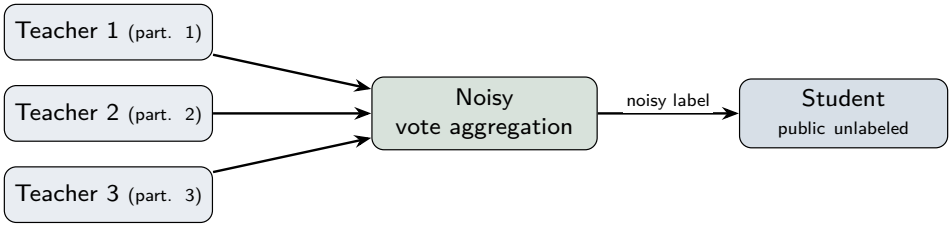
Train (green): DP-SGD and PATE protect the model itself during gradient updates.

Train: DP-SGD recap

- **Stage 2 is what Section 2 covered.** DP-SGD is the canonical Train-stage technique; the green-shaded box on the pipeline diagram marks it as "already covered."
- **In one line.** $\tilde{g}_t = \frac{1}{L}\sum_i \mathrm{clip}(g_t(x_i), C) + \mathcal{N}(0, \sigma^2 C^2 I)$. Three ingredients: clip, noise, accountant.
- **What DP-SGD buys.** A formal (ε, δ)-DP guarantee on the trained model, at a measurable accuracy cost (the accuracy/ε tradeoff slide in §2).
- **Where DP-SGD alone is not enough.** When the model architecture is not a clean fit for per-example gradient clipping (PATE, next slide), or when the model is so large that full DP-SGD is too expensive (DP-LoRA, the Fine-tune stage below).

Train: Private Aggregation of Teacher Ensembles (PATE) for any architecture

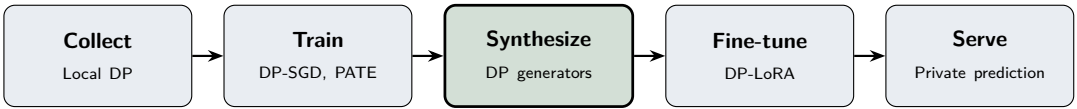
- **Mechanism.** Partition sensitive data → train one teacher per partition → noisy vote-aggregation of teacher predictions → distill into a student trained on public unlabeled data.
- **Why it matters.** Architecture-agnostic; strong privacy from disjoint training; budget spent only when teachers disagree.



Papernot et al. 2018. Disjoint partitions → noisy aggregation → student distillation.

Papernot et al. ICLR 2018; Boenisch et al. PETS 2023 (individualized PATE).

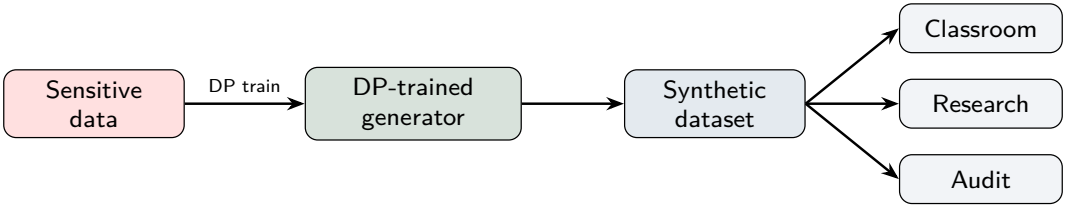
Synthesize



Synthesize (green): a DP-trained generator produces shareable synthetic data with built-in privacy.

Synthesize: DP synthetic data generation

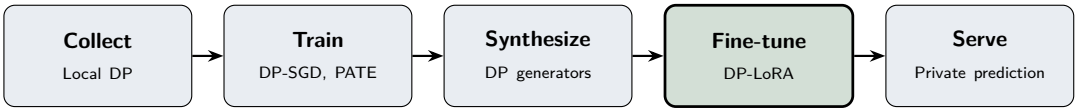
- **Mechanism.** DP-train a generator (VAE / GAN / diffusion / DP-fine-tuned LLM) on sensitive data; release the synthetic dataset; downstream uses inherit the DP guarantee.
- **Why it matters.** Solves "we cannot share the data but we want to enable research and classroom labs."



Synthetic data inherits the DP guarantee by post-processing; downstream uses are unrestricted.

Surveys: *Tabular DP Synthesis* (arXiv 2411.03351, 2024); *How to DP-fy Your Data* (arXiv 2512.03238, 2025).

Synthesize

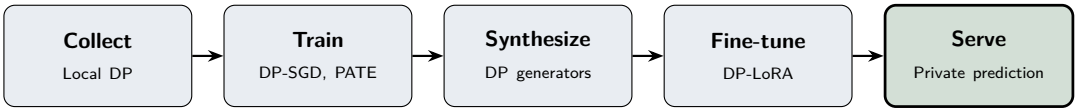


Fine-tune (green): DP-LoRA adapts a frozen base model on private data with bounded leakage.

Fine-tune: DP-LoRA for LLMs

- **Why classical DP-SGD breaks for LLMs.** Full-model DP-SGD costs scale with model size; per-example gradient clipping on a 7B-parameter model is impractical at training scale.
- **DP-Low Rank Adaptation (LoRA).** Apply DP-SGD only to the LoRA adapter weights ($\sim$0.1–1% of full parameters). Memory drops $\sim$33%; training $\sim$8% faster than full DP-SGD.
- **Sources.** Yu et al. *Differentially Private Fine-Tuning of Language Models*, ICLR 2022; ACM Workshop on Privacy in LLMs (2025) empirical evaluation.
- **Example.** Code-completion LLM fine-tuned on private repos is the canonical DP-LoRA setting.

Synthesize



Serve (green): private prediction wraps inference-time queries with DP protections.

Serve: Private prediction at inference time

- **Mechanism.** Add calibrated noise to model outputs (logits or predictions) at query time; track per-query privacy budget.
- **Why it matters.** When the pre-trained model is fixed and you cannot retrain (or do not have permission to retrain), or when fine-grained per-query privacy accounting matters more than a one-shot training budget.
- **Tradeoff with DP-SGD.** DP-SGD pays the privacy cost once during training; private prediction pays per query. The right choice depends on query volume relative to retraining cost.
- **Example.** Vendor serves the code-completion LLM with bounded per-query leakage, without retraining the model each time a privacy policy changes.

Mechanism: output perturbation (Dwork & Roth 2014, §3.3). Per-query budget accounting applies the same composition framework as DP-SGD.

Teach-this-in-your-course: DP-for-ML at three depth tiers

15-min discussion

Show the 5-stage pipeline + DP-SGD recap + one deployed local-DP example (Apple emoji). Faculty leave understanding *where* DP lives.

One-week assignment

Add PATE walkthrough + DP-LoRA reading (Yu et al. 2022) + one homework using Opacus or TF Privacy.

Semester project

Add DP synthetic data lab (TabDDPM or DP-MERF) + cross-stage comparison project + empirical DP auditing reading.

increasing depth, instructor time, classroom payoff

References per tier: Demelius et al. *ACM CSur* 2025 survey (lecture); Yu et al. ICLR 2022 (week); *Tabular DP Synthesis Survey* 2024 (module).

Outline

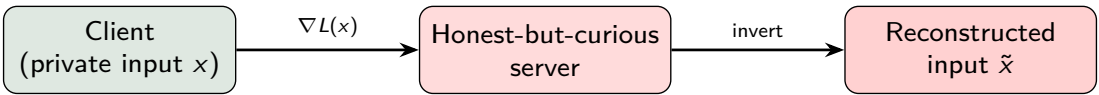
Federated Learning

What is it?

- Distributed training across clients that cannot share data.
- Clients compute local updates and send them to a central server for aggregation.
- Server aggregates the updates to improve the global model and sends the updated model back to clients.
- Repeats for multiple rounds until convergence.

Threat-first: gradient inversion on Multi-enterprise Phishing Filter

- **Setup.** Example B – multi-enterprise phishing classifier, federated across enterprises that cannot share inboxes.
- **Attack.** *Gradient inversion* – reconstructing the training input from a single gradient update (Geiping et al. 2020).
- **Evaluation.** Huang et al. 2021 tempers the worst-case claim; GI is real but harder than the headline.
- **Implication.** FL is decentralization, not privacy. Composition with DP and/or secure aggregation is required.



Geiping et al. 2020; Huang et al. 2021 evaluation tempers worst-case claims.

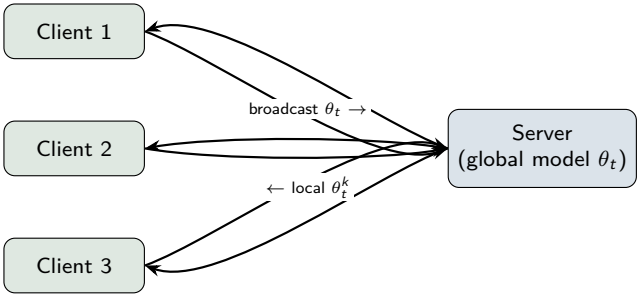
FedAvg: the basic structural defense

Each round the server computes a weighted average of client updates:

$$\theta_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \theta_t^k$$

Data stays on clients; only model updates move.

McMahan et al. AISTATS 2017; Cha et al. 2025 multi-level aggregation taxonomy.



What FL leaks on its own

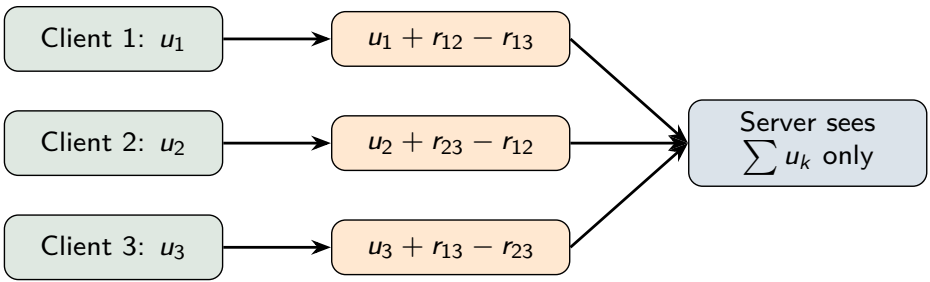
Even with no centralized data, FL leaks via:

- **Gradient inversion.** Per-round gradient updates can be inverted to recover training inputs.
- **Model-update inversion.** Multi-round θ_t^k inspection reveals more than any single round.
- **Membership inference at the aggregator.** Same attack family, now run by the server.

Implication. FL needs composition (secure aggregation, DP, or both) to become a privacy defense. *Secure aggregation, next slide, closes the per-update inspection channel.*

Secure aggregation: combining updates without seeing them

- **Mechanism.** Clients mask updates with pairwise random masks; masks cancel in the sum but not in any individual update.
- **Server's view.** The sum $\sum_k u_k$ (which it needs for FedAvg) – not any individual u_k.
- Bonawitz et al. CCS 2017. Recent review: Wang et al. *Future Internet* 2025.



Bonawitz et al. 2017. Pairwise masks cancel in the sum.

Malicious clients: Prio and VDAFs

Secure aggregation tolerates an honest-but-curious *server*. It does not by itself tolerate malicious *clients* who submit malformed updates.

- **Prio** (Corrigan-Gibbs & Boneh, NSDI 2017). Zero-knowledge proofs on secret-shared data verify that each contribution stays within bounds.
- **VDAFs** (IRTF/CFRG draft). Generalize Prio for verifiable distributed aggregation; moving toward standardization.
- **Frontier.** Roy Chowdhury et al. *eprint 2023/130*; Carletti et al. USENIX Sec 2025 *SoK on Gradient Inversion*.

Faculty pointer: `dpsa4fl` open-source project combines VDAF-based secure aggregation with DP.

Deployed FL examples

- **Gboard (Google).** Next-word prediction trained across millions of phones; FedAvg + server-side DP.
- **Apple on-device learning.** Keyboard, photo, and speech models with a similar FL + server-side DP architecture.
- **Cross-hospital medical imaging.** Consortium training where institutions cannot pool raw images (Ziller et al. 2021 numbers as a benchmark).
- **Tool pointer.** `Flower` – open-source FL framework with classroom-friendly tutorials.

Teach-this-in-your-course: FL at three depth tiers

15-min discussion

FedAvg in 20 minutes + one deployed example (Gboard) + one "what does FL leak?" discussion.

One-week assignment

Add secure aggregation + Flower-based simulation lab + one reading (Wang et al. 2025 SA review).

Semester project

Add Prio/VDAFs + malicious-clients discussion + project (extend Flower with verifiable aggregation).

increasing depth, instructor time, classroom payoff

Outline

Threat-first: untrusted hosted inference on Example C

- **Setup.** Example C – code-completion LLM fine-tuned by your organization but *hosted by a vendor*. Every query (every proprietary code snippet) crosses the trust boundary.
- **Constraint.** Protect queries at inference time without revealing them to the vendor.
- **Answer family.** *Secure inference* under MPC or homomorphic encryption.
- **Foundational triad.** Kumar et al. CrypTFlow (S&P 2020); Mishra et al. Delphi (USENIX Sec 2020); Knott et al. CrypTen (NeurIPS 2021).

Secure inference also applies to: a shared classifier serving multiple enterprises (Example B); multi-jurisdiction tabular models (Example A).

Secure inference vs. secure training

Secure inference

- Client sends encrypted query; server computes on ciphertext; returns encrypted result.
- Currently practical for moderate-size models.
- HE in one line: $\mathrm{Enc}(x) \oplus \mathrm{Enc}(y) = \mathrm{Enc}(x + y)$ – compute on encrypted data without decrypting.
- Foundational: CrypTFlow (S&P 2020), Delphi (USENIX Sec 2020).

Secure training

- Multi-party training under MPC – training data split across distrusting parties.
- Still expensive enough that real deployments are rare.
- Foundational: Mohassel & Zhang *SecureML* (S&P 2017).

Recent surveys: Yang et al. *ACM TKDD* 2025; Naresh & Reddy *WIREs* 2025 (healthcare MPC).

Cost orders: what is feasible today

- **Secure inference (MPC).** $\sim$10–100$\times$ plaintext for small CNN models.
- **Secure inference (HE).** $\sim$100–10,000$\times$ plaintext for the same models.
- **Secure training.** Rarely deployed; existing systems handle small models on small datasets.
- **When to reach for PPML.** Model itself is sensitive, *or* training data lives across distrusting parties who cannot pool it.

Zhang et al. *ACM CSur* 2025 systematic review of PPML efficiency.

Teach-this-in-your-course: PPML at three depth tiers

15-min discussion

Secure inference vs. training in 20 minutes + Naresh *WIREs* 2025 healthcare reading + "when is the cost worth it?" discussion.

One-week assignment

Add HE additive property + paper-reading lab (Yang *ACM TKDD* 2025 + one specific system paper).

Semester project

Add MPC fundamentals (secret sharing, garbled circuits at concept level) + Crypto-PPML Workshop pointer + research project.

increasing depth, instructor time, classroom payoff

Outline

The consolidated 3×3 menu

	One lecture	One week	One module
DP	ε-DP definition + one deployed ε value	Add (ε, δ), sensitivity, Gaussian, DP-SGD lab	Add composition, Rényi DP, extended project
FL	FedAvg + Gboard + "what FL leaks"	Add secure aggregation + Flower lab	Add Prio/VDAFs + contest-style project
PPML	Secure inference vs. training + Naresh reading	Add HE property + paper-reading lab	Add MPC fundamentals + workshop pointer

Key References

Full annotated list: `topics/09-Privacy-in-AI-ML/resources.md`.

Attacks and evaluation

- Carlini et al. *Extracting Training Data from LLMs*, USENIX Sec. 2021. arxiv.org/abs/2012.07805
- Carlini et al. *Membership Inference Attacks From First Principles*, S&P 2022. arxiv.org/abs/2112.03570
- Shokri et al. *Membership Inference Attacks Against ML Models*, S&P 2017. arxiv.org/abs/1610.05820
- Geiping et al. *Inverting Gradients*, NeurIPS 2020. arxiv.org/abs/2003.14053
- Huang et al. *Evaluating Gradient Inversion*, NeurIPS 2021. arxiv.org/abs/2111.04706

Differential privacy

- Dwork & Roth. *The Algorithmic Foundations of DP*, 2014. cis.upenn.edu/~aaroth/privacybook.pdf
- Abadi et al. *DP-SGD*, CCS 2016. arxiv.org/abs/1607.00133
- Papernot et al. *PATE*, ICLR 2018. arxiv.org/abs/1802.08908

Federated learning + secure computation

- McMahan et al. *FedAvg*, AISTATS 2017. arxiv.org/abs/1602.05629
- Bonawitz et al. *Practical Secure Aggregation*, CCS 2017. eprint.iacr.org/2017/281
- Mohassel & Zhang. *SecureML*, S&P 2017. eprint.iacr.org/2017/396
- Kumar et al. *CrypTFlow*, S&P 2020. arxiv.org/abs/1909.07814
- Mishra et al. *Delphi*, USENIX Sec. 2020. usenix.org/.../mishra
- Knott et al. *CrypTen*, NeurIPS 2021. arxiv.org/abs/2109.00984

Legal / policy / unlearning

- Bourtoule et al. *Machine Unlearning (SISA)*, S&P 2021. arxiv.org/abs/1912.03817
- Chen et al. *When Machine Unlearning Jeopardizes Privacy*, CCS 2021. arxiv.org/abs/2005.02205
- NIST AI RMF v1.0, 2023. nist.gov/itl/ai-risk-management-framework