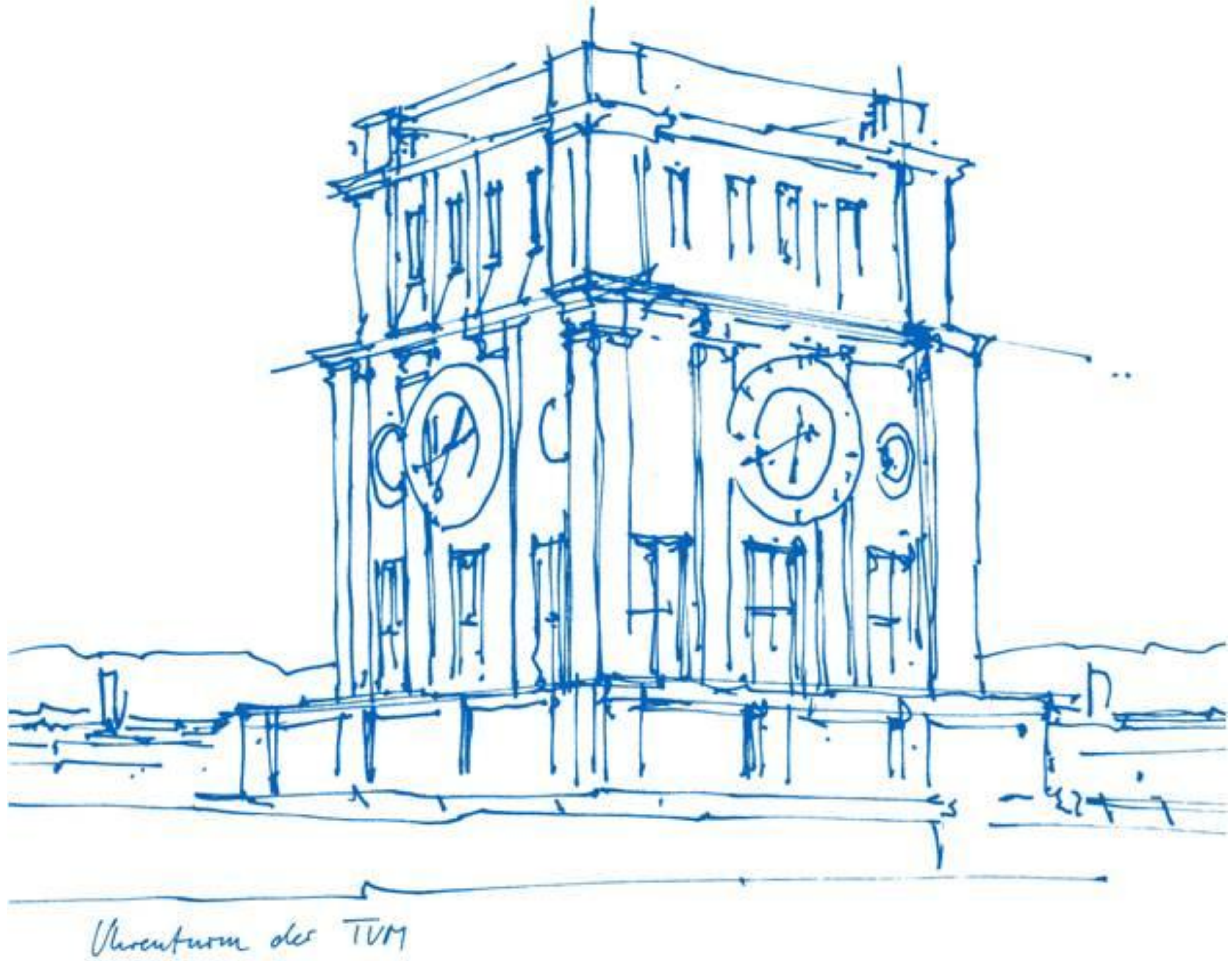
Adversarial Interference in Collaborative Machine Learning

Talk by: *Dmitrii Usynin*

3rd year PhD @Imperial College London / Technical University of Munich

(incoming) Machine Learning Researcher @Brave



In this talk we concentrate on privacy and robustness of ML models

Privacy:

- Concerns both the *input* and the *output* privacy
- *Input* - data cannot be seen by an unauthorised party
- *Output* - the results of the computation do not reveal sensitive information

Robustness:

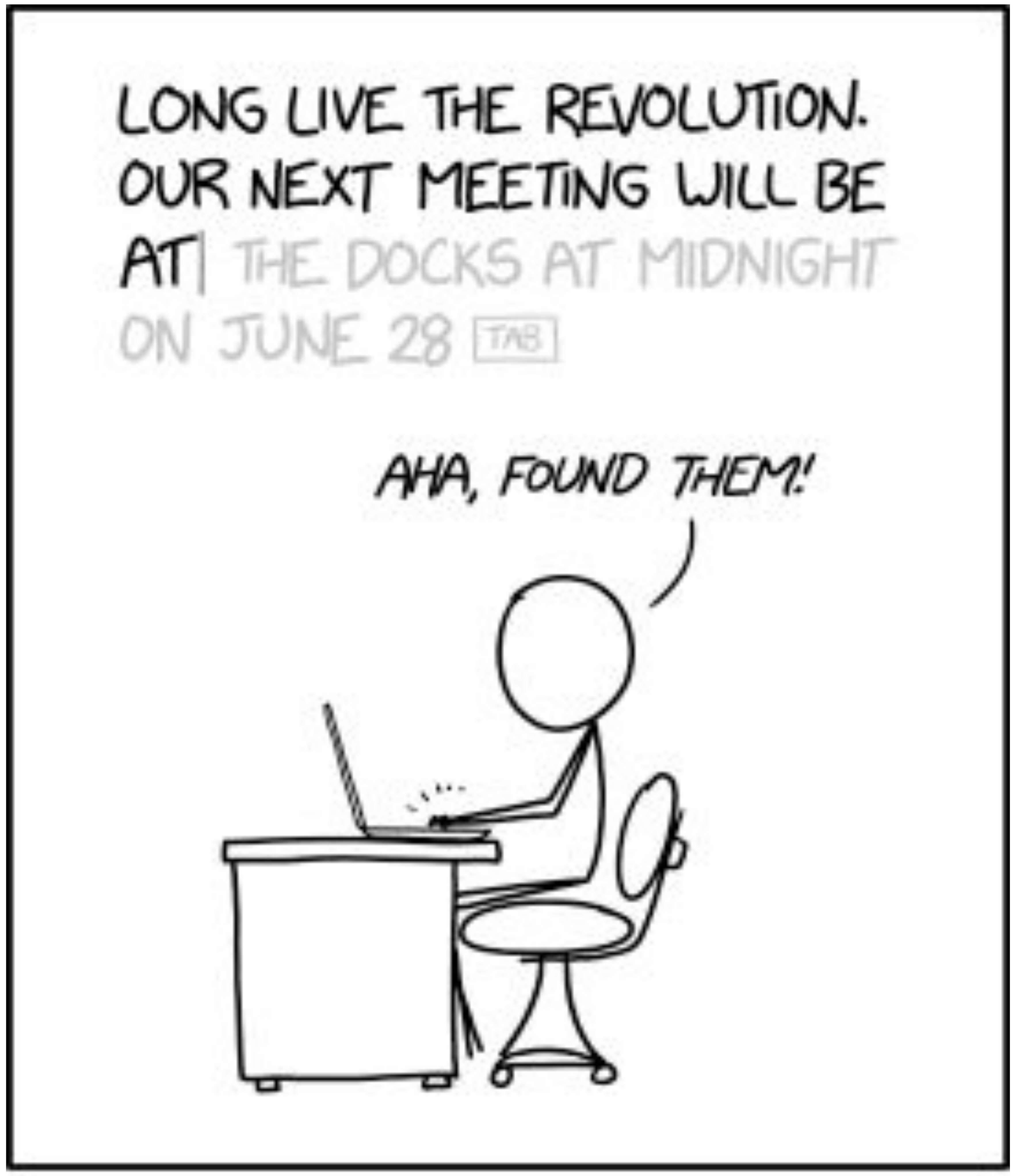
- Concerns *verification* and *accountability*
- *Verification* - model behaves as intended when trained
- *Accountability* - all contributions to the training algorithm can be linked to the individual parties

Why bother with privacy?

Unintended (or not) memorization can occur in many sensitive machine learning contexts:

- Biomedical (genetic data)
- Financial (credit records)

As of 2021, this cartoon is no longer an exaggeration



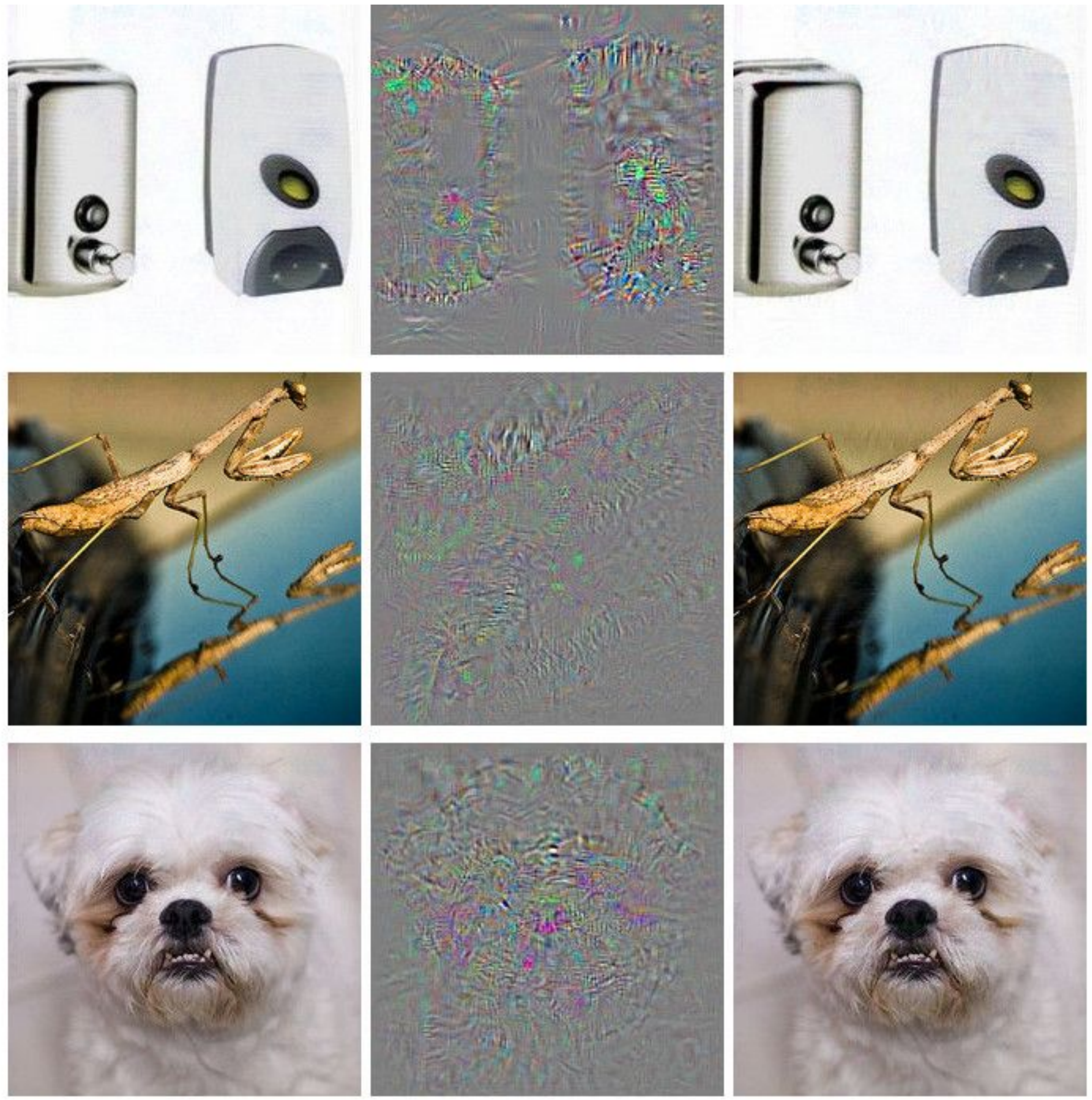
WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

Why bother with robustness?

Most collaborative training paradigms rely on collaborators sharing model updates and not the training data itself:

- Difficult to verify integrity
- In certain cases contributions are also anonymous

What unites these images?



Overview of attacks

Privacy-Centred attacks:

- Attempt to disclose <u>information</u> participants <u>did not consent to disclosing</u>
- Examples include: *membership, sensitive attributes, training records reconstruction etc.*

Utility-Centred attacks:

- Attempt to subvert the protocol and <u>alter the utility</u> of the model
- Examples include: *crafting malicious data or updates, hidden collateral tasks etc.*

(Brief) overview of defenses

Privacy-Centred attacks:
(e.g. model inversion)

- Secure multi-party computation
- Homomorphic encryption
- Trusted execution environments
- Differential privacy

Utility-Centred attacks:
(e.g model poisoning)

- Model pruning
- Knowledge distillation
- Adversarial training
- Robust aggregation
- Update tracking and analysis

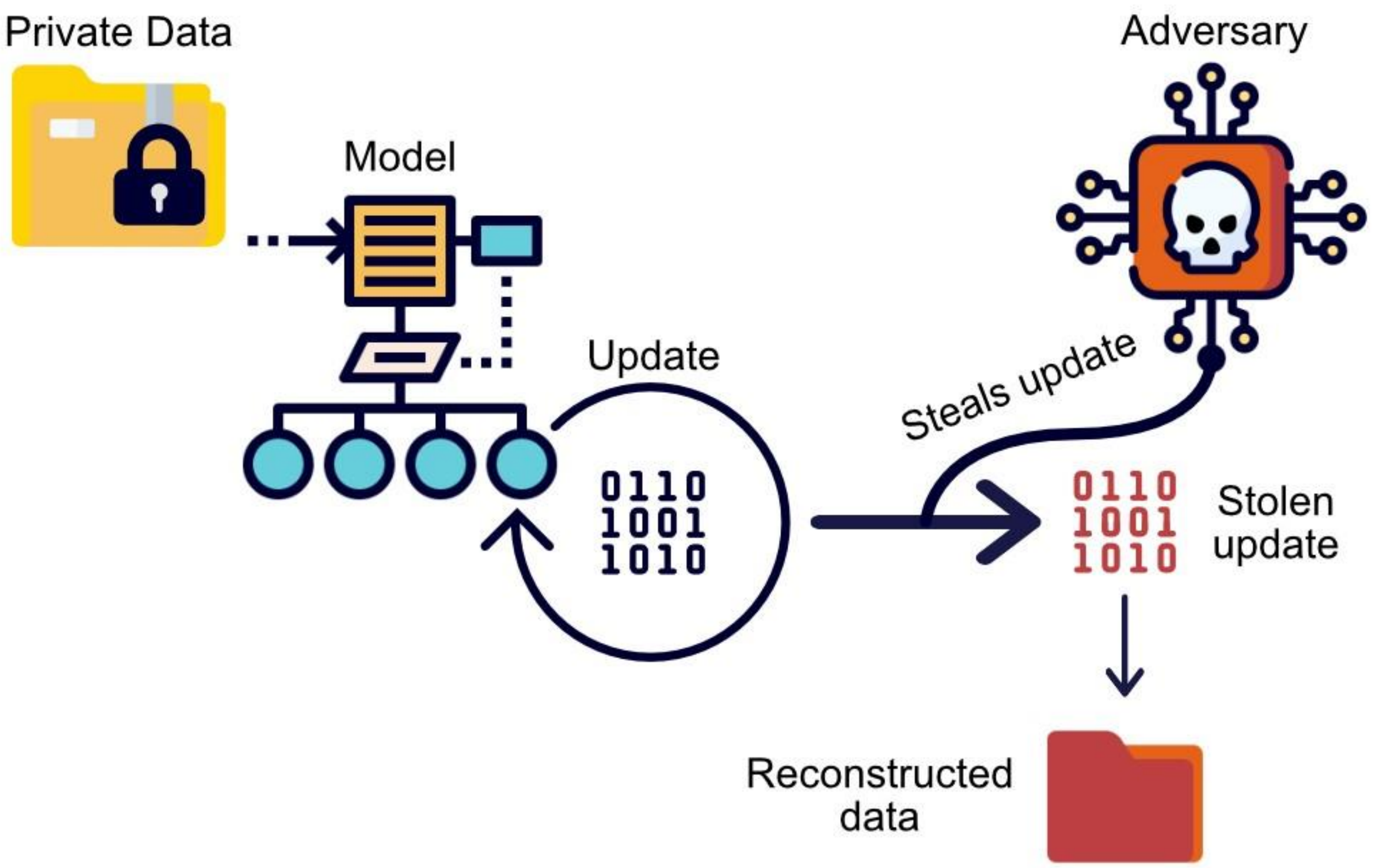
N.B. Some of these methods are incompatible with each other

Overview of existing attacks

Model inversion

Attacker uses internal representations of the joint model to reconstruct individual training samples or their sensitive attributes

Example: inversion of training data in collaborative pneumonia classification

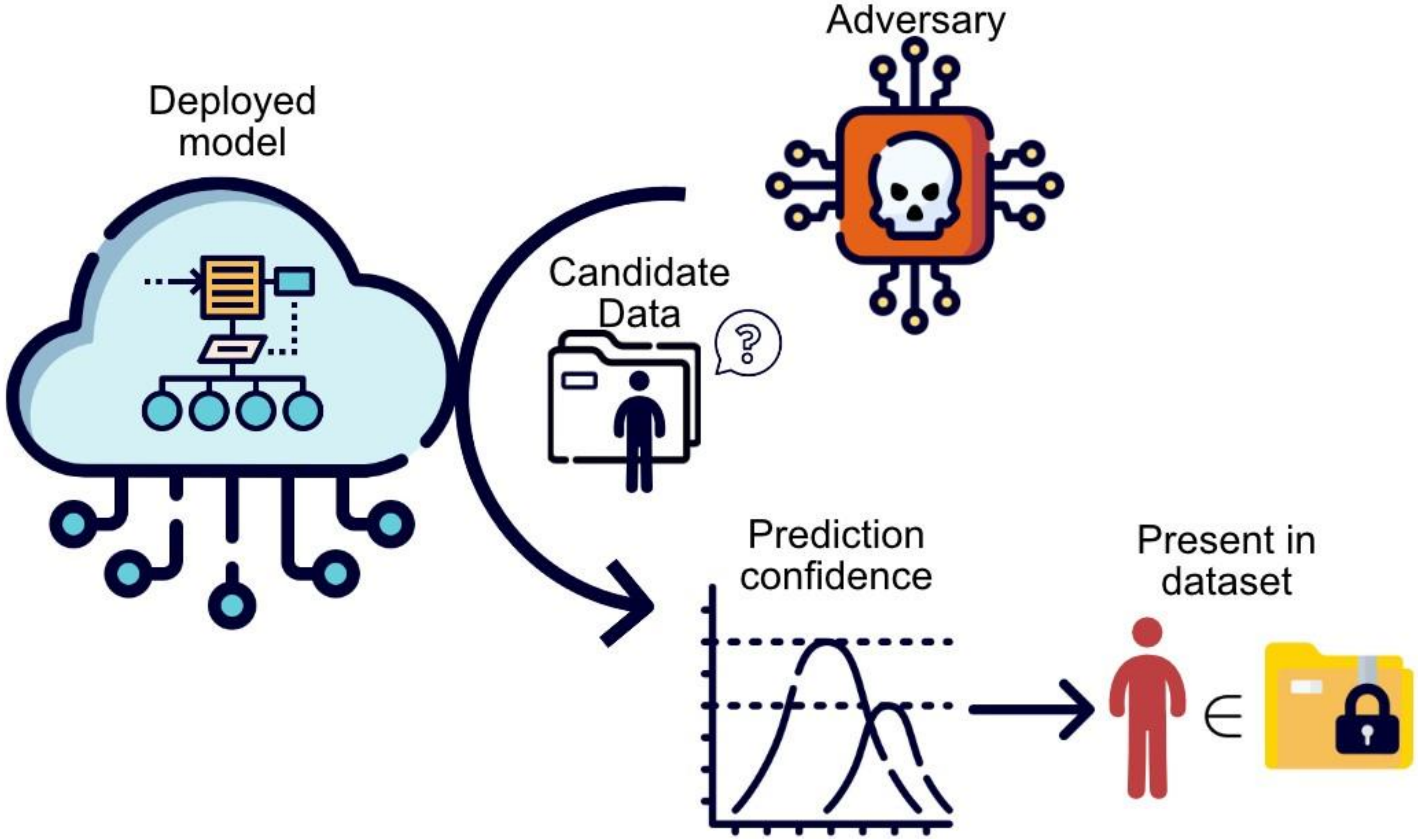


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Membership inference

Attacker obtains a data record and determines if it was used to train a particular model

Example: determining if a specific patient was part of the HIV-positive dataset

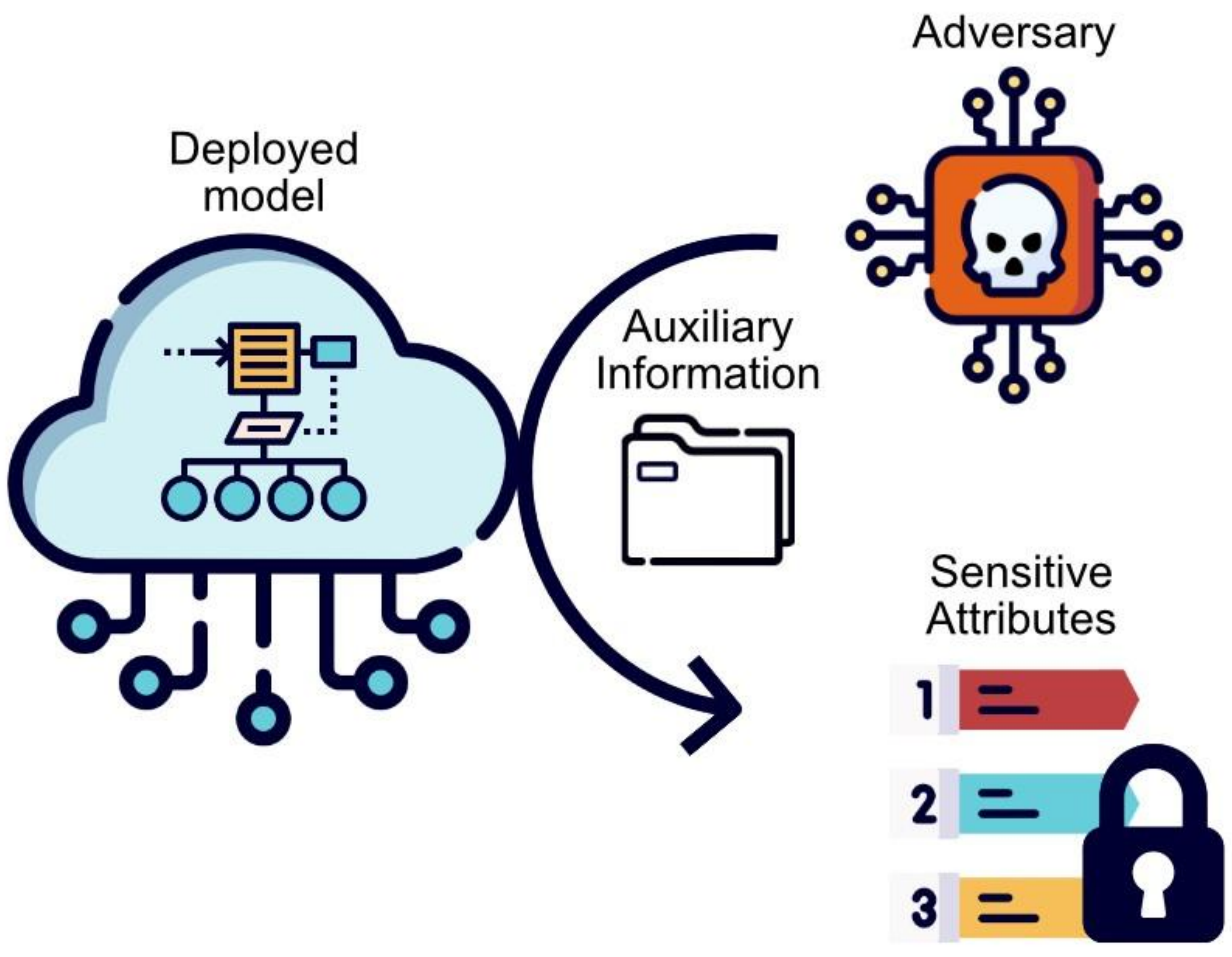


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Attribute inference

Attacker uses model access and auxiliary information about the victim to obtain the sensitive values of their data

Example: given access to a model trained on patient records and a specific patient's public information infer their HIV status

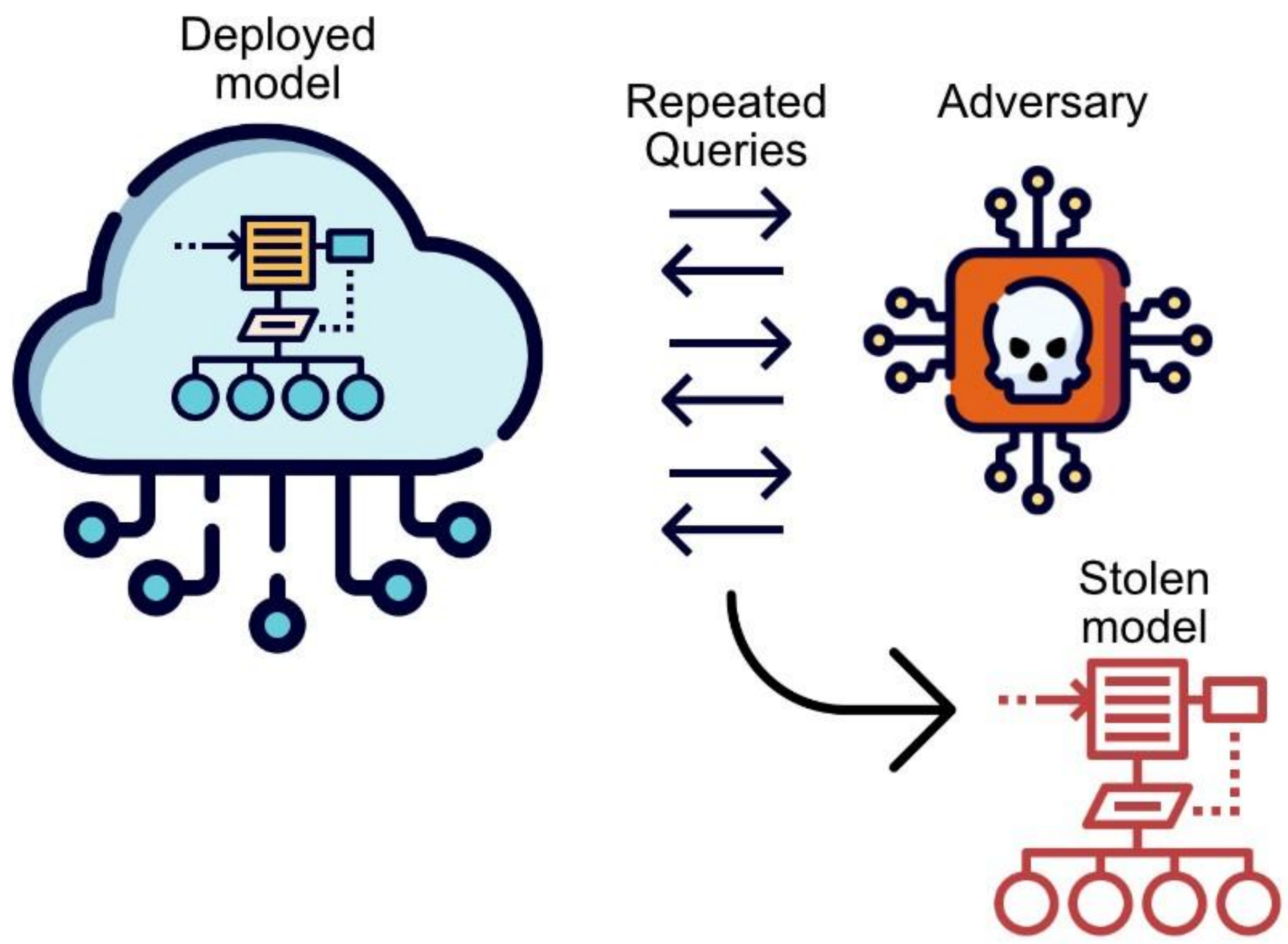


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Model extraction

Attacker only has black-box access to the model and obtains its in entirety, architecture or parameters via queries

Example: querying a proprietary MLaaS deployed model to approximate its parameters

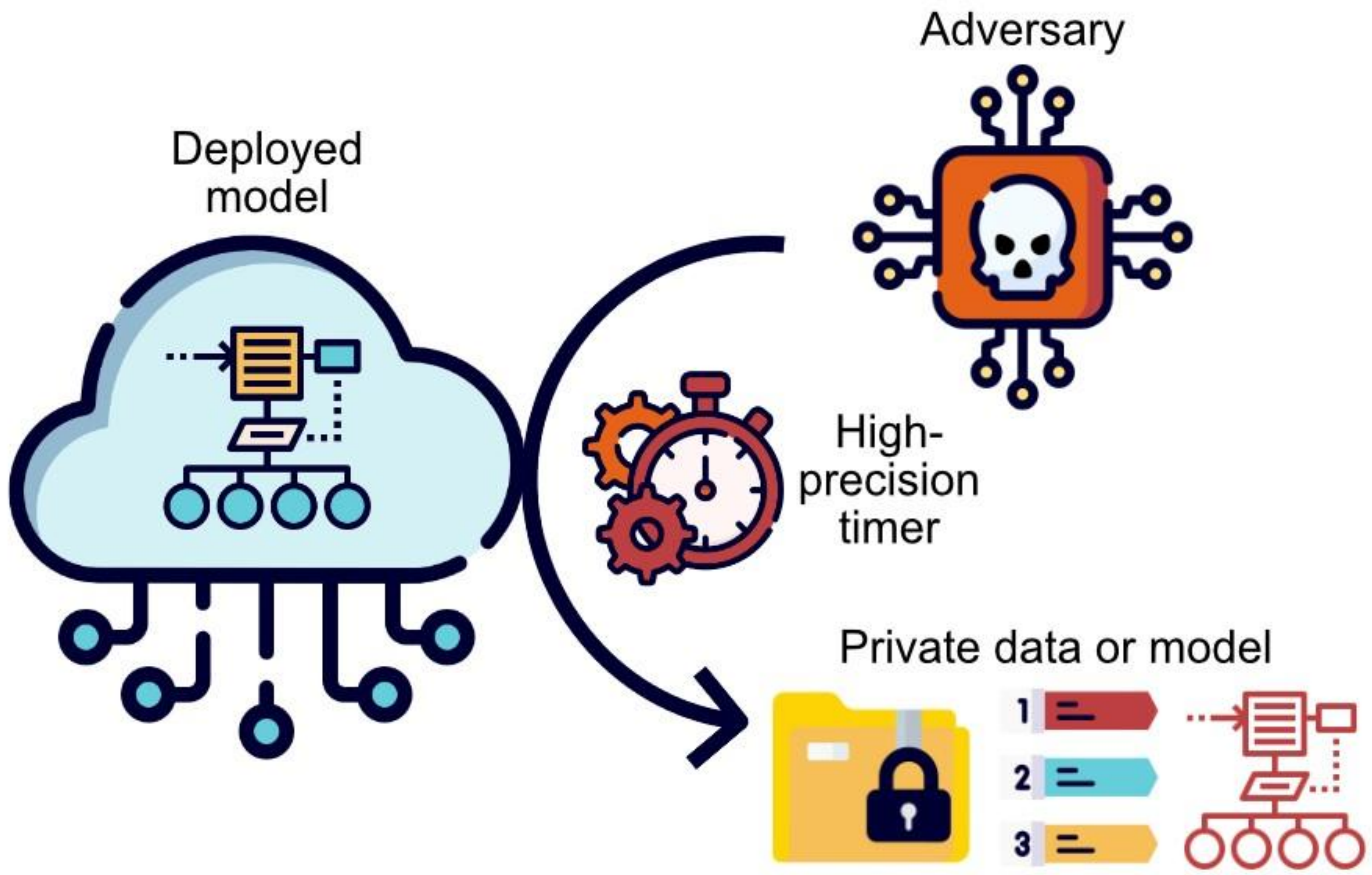


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Side-channel attack

Attacker utilises unrelated information about the victim/target system to obtain data they can use to extract sensitive information

Example: using physical machine access to analyse the individual operations to determine model architecture

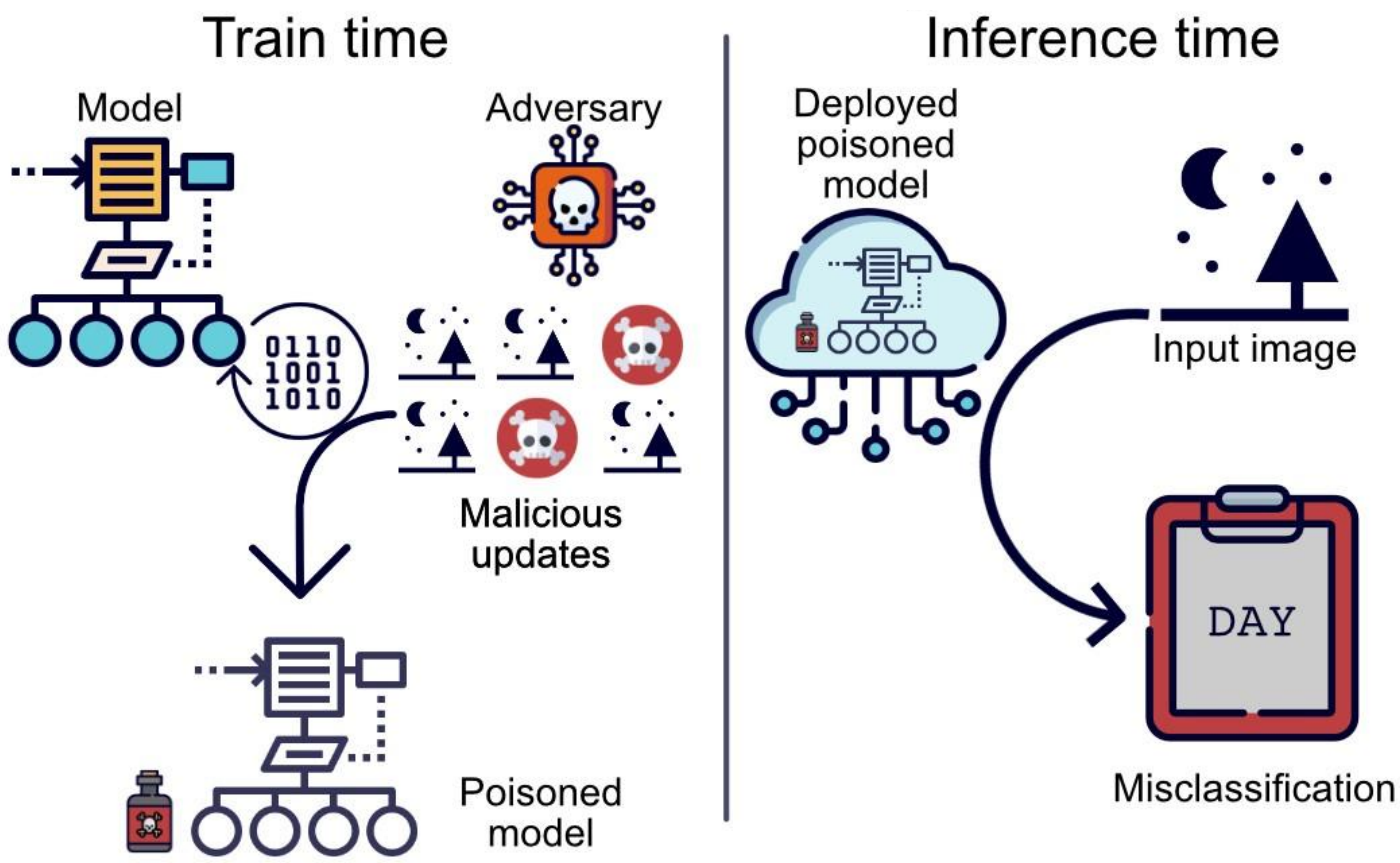


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Model poisoning

Attacker submits updates aimed at reducing the utility of the model via input perturbations

Example: perturbation of chest X-rays in collaborative pneumonia classification to reduce the accuracy of the final model

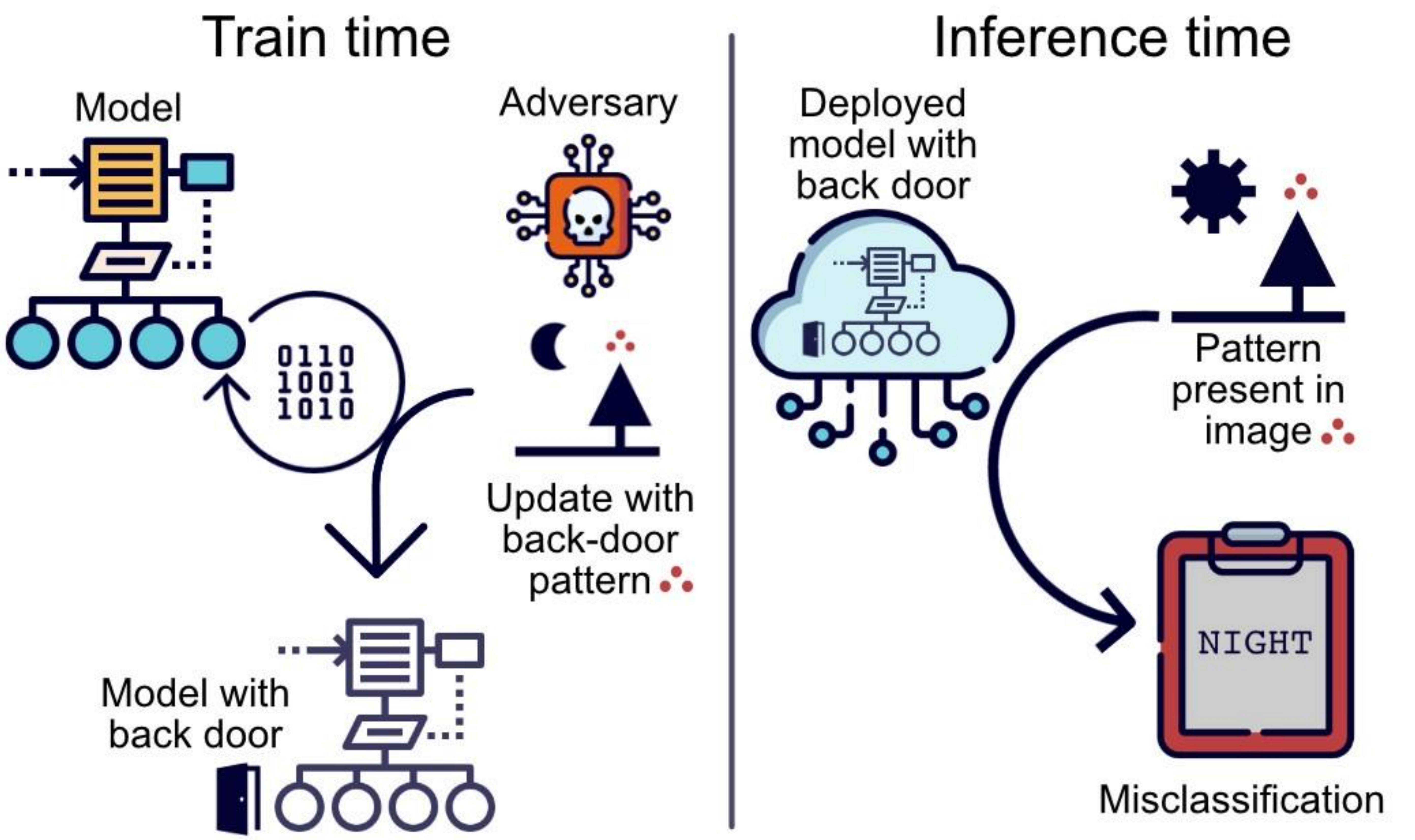


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Backdoor attack

Attacker modifies the model to contain a hidden learning task that either benefits the attacker or destroys utility for a specific subgroup

Example: embed a hidden pattern in images used in collaborative pneumonia classification to reduce the accuracy for black patients

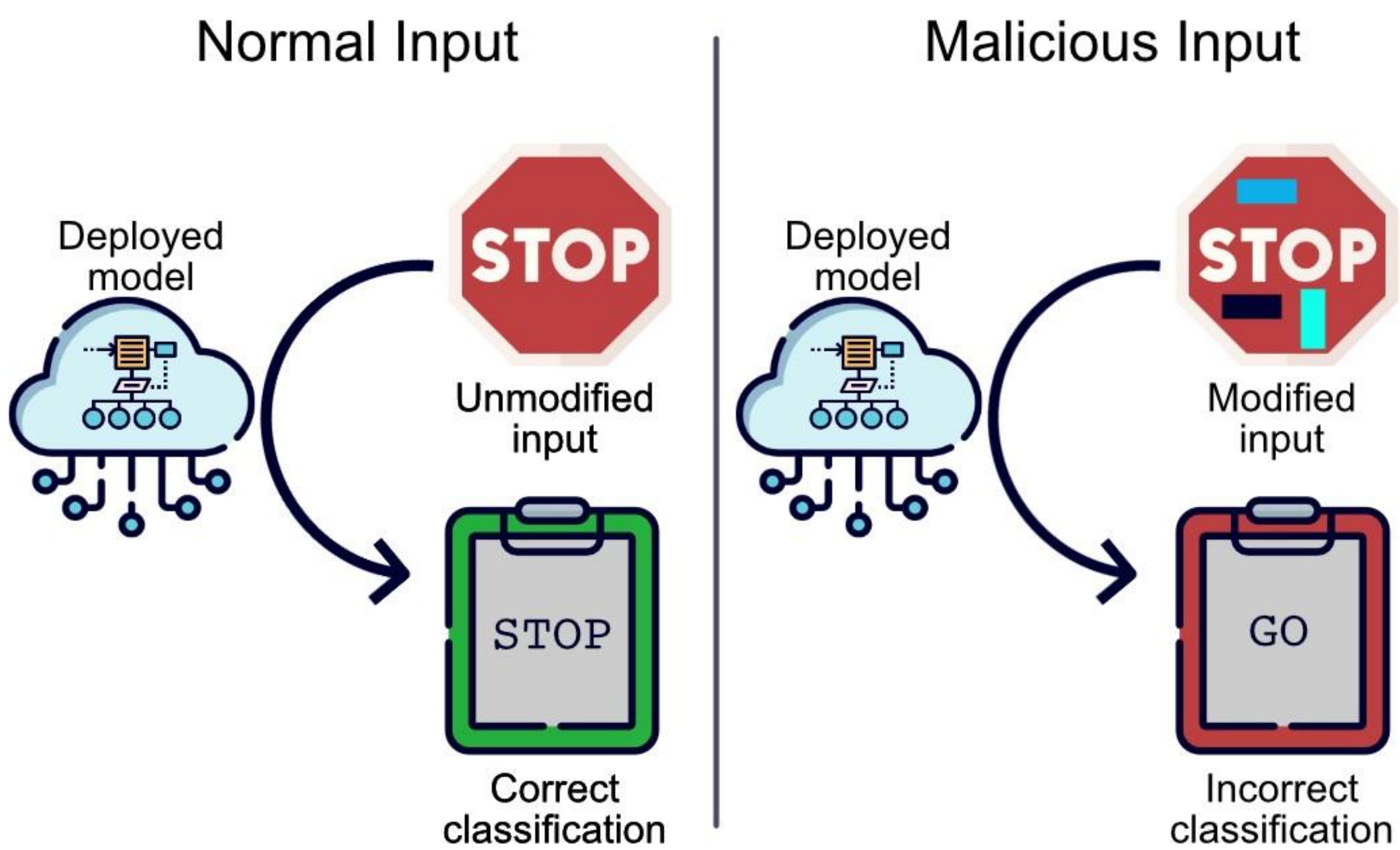


Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

Evasion attack

Attacker modifies inputs at test time to force the model to mispredict

Example: putting stickers on the road signs in an area where fully-autonomous vehicles are deployed



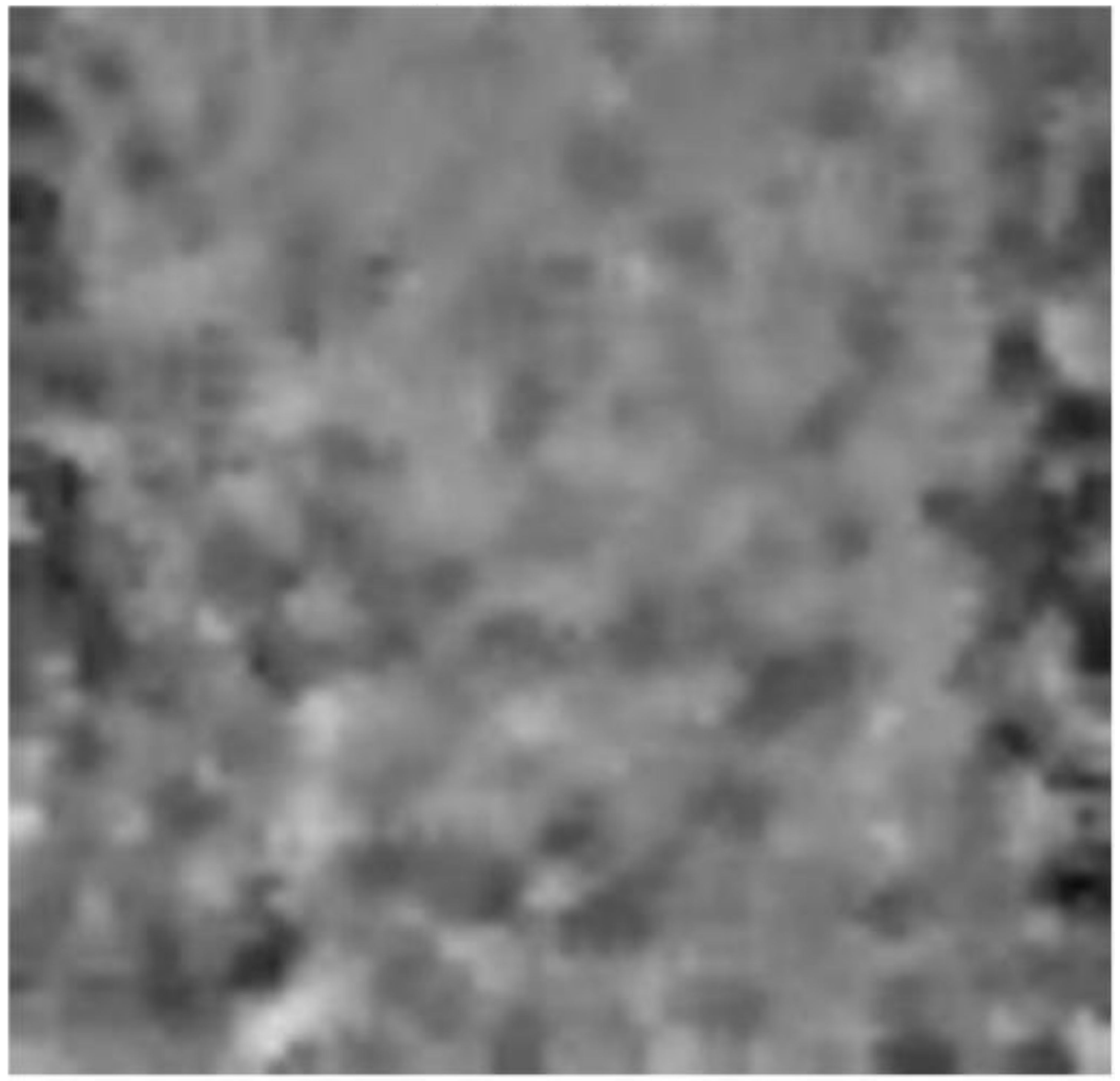
Usynin et al. 2021, Adversarial interference and its mitigations in privacy-preserving collaborative machine learning, *Nature Machine Intelligence*

What is being done to prevent this?

	Overview	Advantages	Disadvantages	References
Privacy-centred defences				
Homomorphic Encryption	Training is performed on encrypted data only decrypting the final output	- Formal guarantees - Secure by design - Several schemes available	- Computational overhead - Function approximations reduce model utility - Susceptible to utility and membership attacks	[80–82]
Secure Multi-Party Computation	Compute shared function without revealing inputs to other clients	- Formal guarantees - Secure by design - Many implementations available	- Communication overhead - Susceptible to utility and membership attacks	[83–85]
Trusted Execution Environments	Run (part of) training on secure enclave	- Formal guarantees - Secure by design	- Additional hardware requirements - GPU training nascent - Susceptible to side-channel attacks	[86, 87]
Knowledge Distillation	Transfer of knowledge from a public model to the private one	- Prevents overfitting - Scalable - No computation overhead	- Requires publicly available dataset - No formal guarantees	[58, 60, 61]
Split Learning	Model is trained locally up to a cut layer, the rest is trained on the other host	- Scalable - Reduced communication overhead compared to FL	- Susceptible to reconstruction attacks - Susceptible to utility and membership attacks	[69, 88]
Utility-centred defences				
Data Analysis	Analyse data from other clients and perform pre-processing if required	- Empirically effective	- Violates privacy - Difficult to execute under data protection regulations	[71–73]
Update Analysis	Analyse updates from other clients and determine if they should be aggregated	- Flexible metrics for updates - Empirically effective	- Violates privacy - Not effective against backdoors	[30, 70, 72]
Robust Aggregation	Replace update averaging with an aggregation based on utility and/or data analysis	- Allows more efficient training - Empirically effective	- No formal guarantees - Can reduce model utility - Some updates are discarded thus wasting computation resources - Susceptible to privacy attacks	[28, 66, 89]
Adversarial Training	Train the model on adversarially crafted samples in addition to regular ones	- Empirically effective - Easy to implement - No computation overhead	- No formal guarantees - Has a small impact on model utility - Susceptible to privacy attacks	[56, 71, 90, 91]
Shared defences				
Model Pruning	Discard specific neurons/units of the model based on a pre-defined strategy	- Large number of implementations - Prevents overfitting - Easy to include in training	- Often ineffective - Can reduce model utility	[64]
Differential Privacy	Targeted perturbation of certain stages of the protocol to make the algorithm approximately invariant to addition/exclusion of data points	- Formal guarantees - Implicit regularisation - Improved robustness - Scalable to many parties	- Reduced model utility - Increased training time - Not formally effective against utility attacks	[91–94]

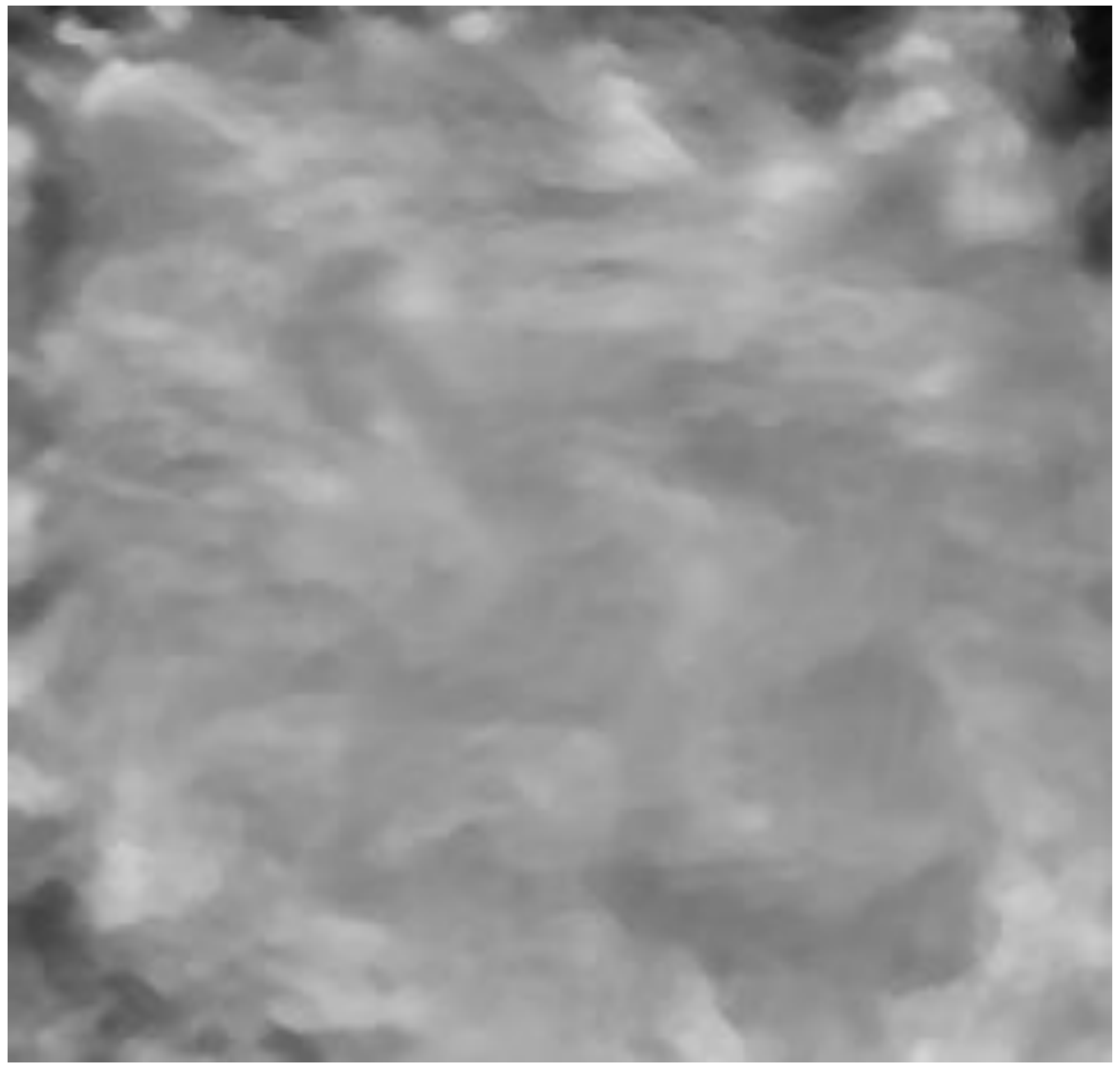
Some defenses are theoretical and some are empirical

Reconstruction attack on a DP-trained model



Usynin et al., 2021, Zen and the art of model adaptation: Low-utility-cost attack mitigations in collaborative machine learning, *PETS Symposium 2022.1*

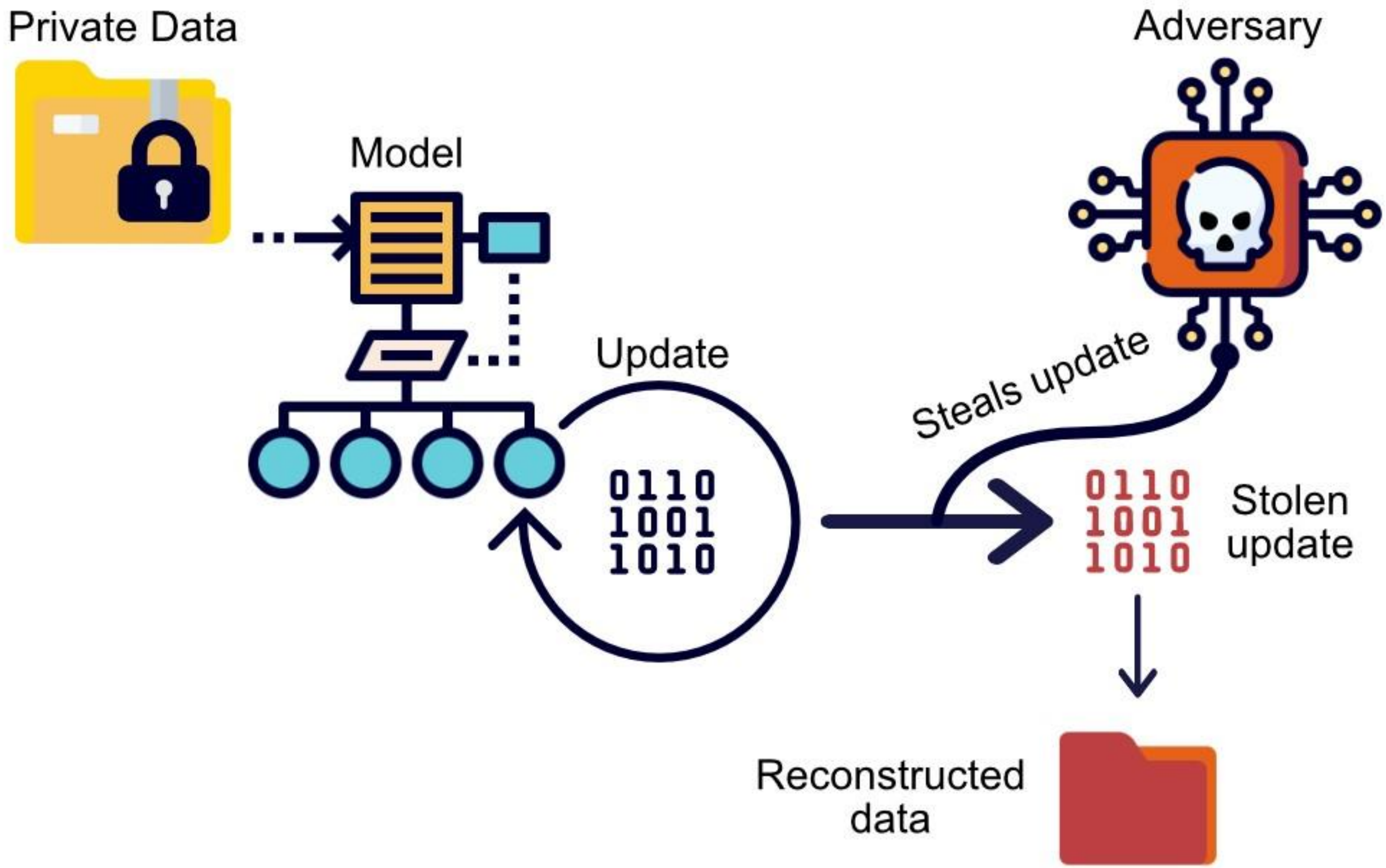
Reconstruction attack on a model trained with *model adaptations*



So in most cases there no 'standard' defense that can protect your settings against all adversaries

Concrete attack-mitigation example

Concrete attack example: Gradient-based model inversion



Concrete attack example: Gradient-based model inversion

1. The adversary randomly generates an image-gradient pair

2. The adversary captures the gradient update submitted by the victim

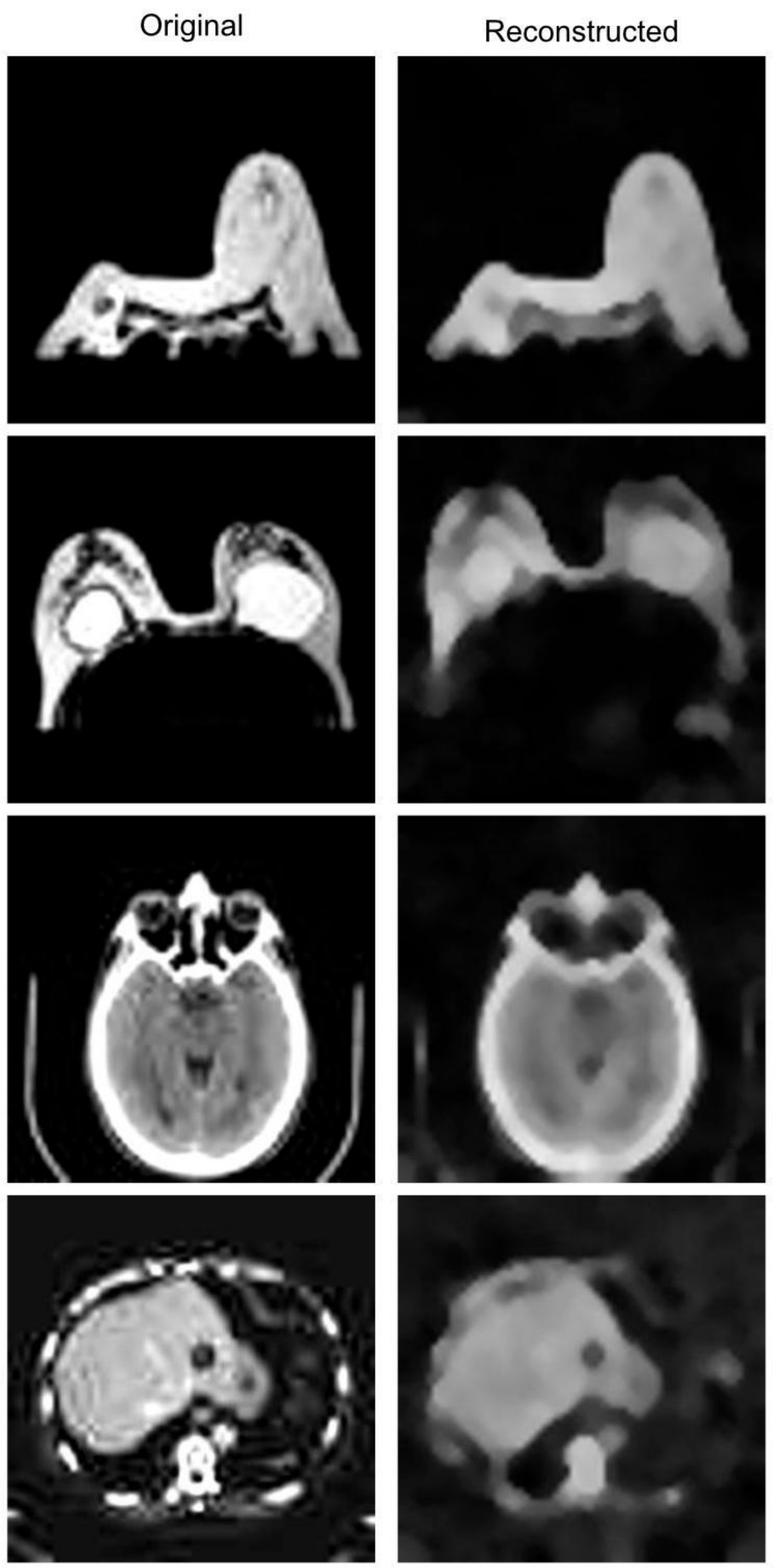
3. Using a suitable cost function (often cosine similarity), the adversary minimises the difference between the captured and the generated updates by perturbing the image they control

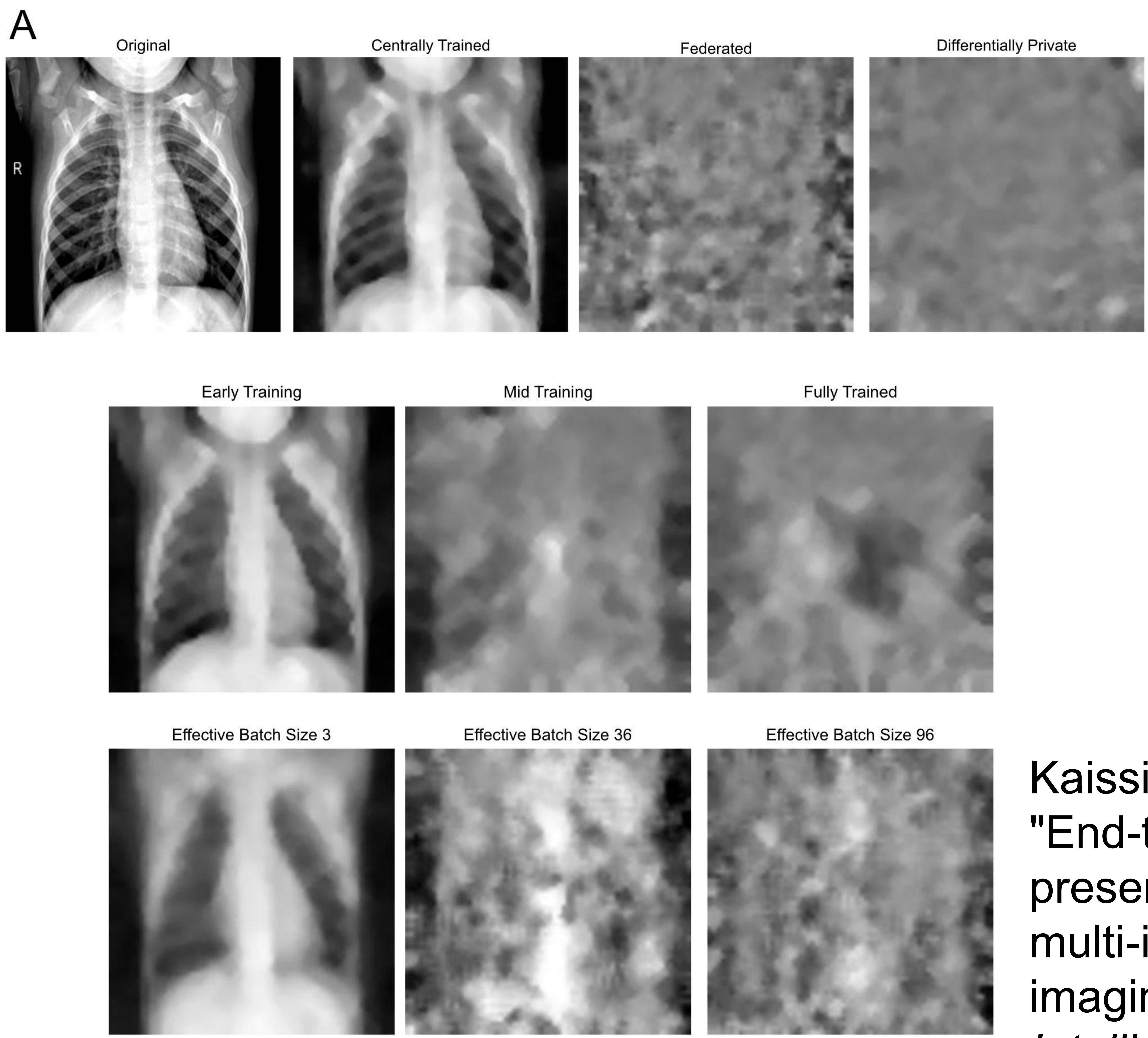
4. The algorithm is repeated until the final iteration is reached.

$$\arg \min_{x' \in [0,1]^n} \left\{ 1 - \frac{\langle \nabla_\theta \mathcal{L}(x, y), \nabla_\theta \mathcal{L}(x', y) \rangle}{\|\nabla_\theta \mathcal{L}(x, y)\|_2 \cdot \|\nabla_\theta \mathcal{L}(x', y)\|_2} + \alpha \mathrm{TV}(x) \right\}$$

where x' is the reconstruction target, x is the ground truth, y is the label, $\nabla_\theta \mathcal{L}$ is the gradient with respect to the weights, $\langle \cdot \rangle$ is the inner product in $\mathbb{R}^n$ and $\| \cdot \|_2$ is the L_2-norm. α is a hyperparameter scaling the total variation penalty over the image, $\mathrm{TV}(x)$.

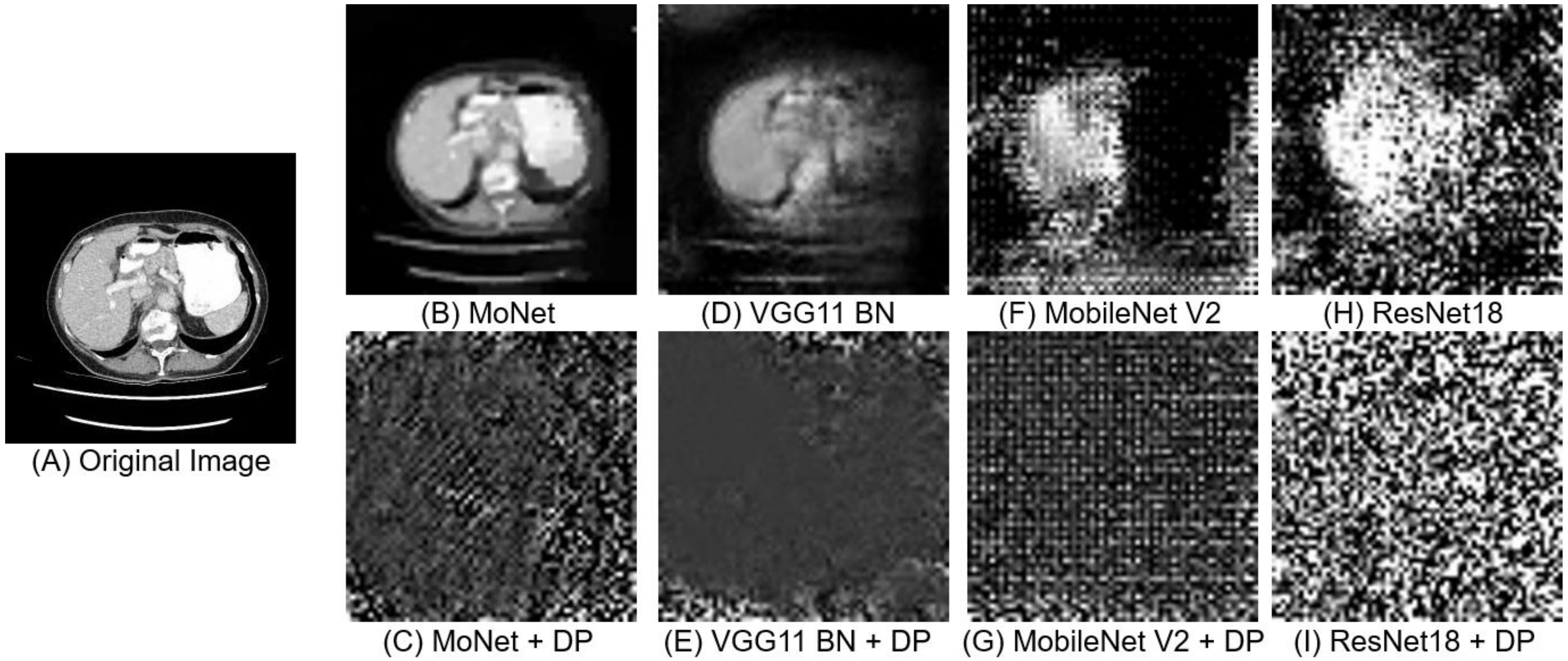
Examples of gradient inversion in practice (classification):



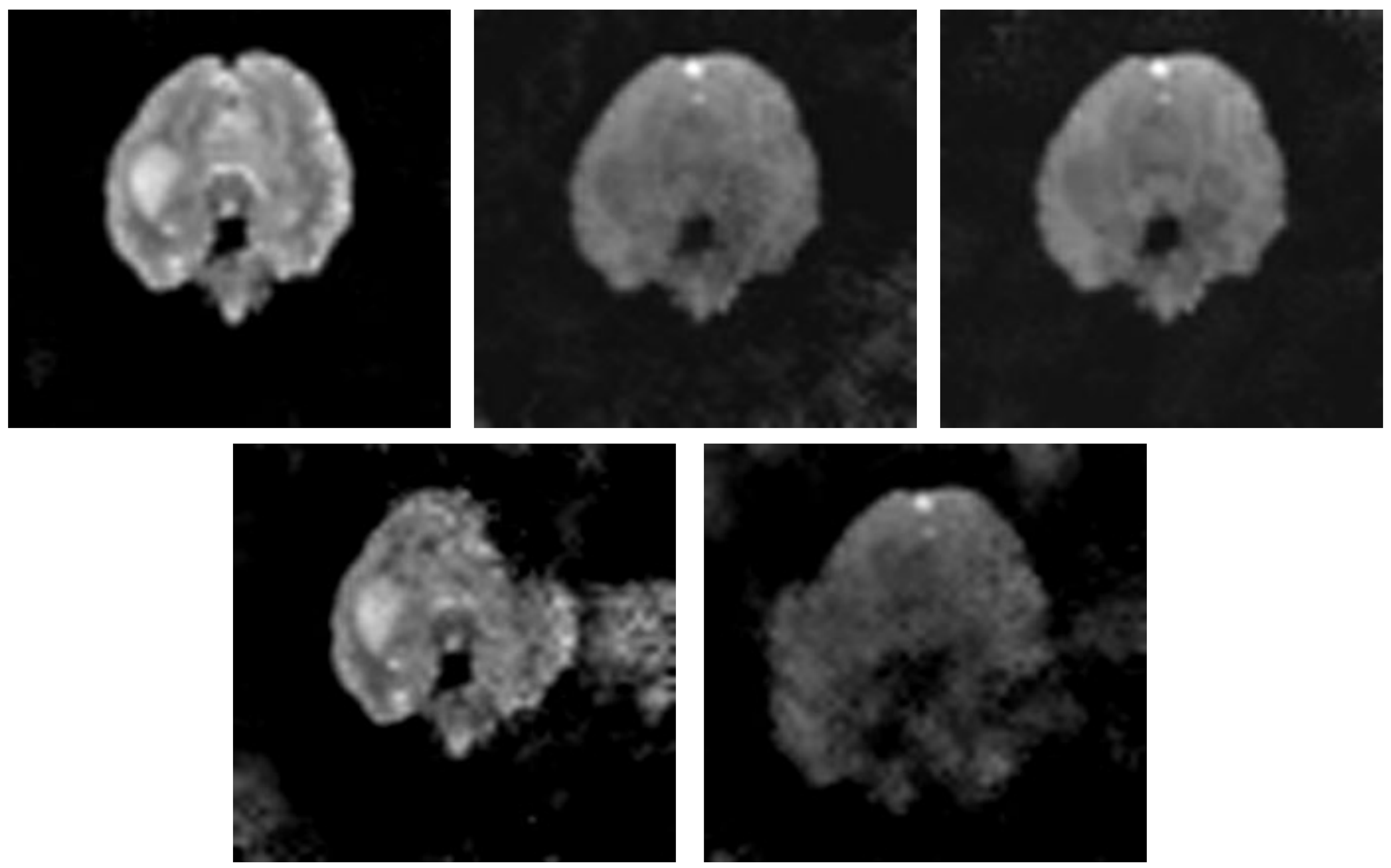


Kaissis, Ziller, et al. "End-to-end privacy preserving deep learning on multi-institutional medical imaging." *Nature Machine Intelligence* 3.6 (2021): 473-484.

Examples of gradient inversion in practice (segmentation):



Ziller et al. "Differentially private federated deep learning for multi-site medical image segmentation." *arXiv preprint arXiv:2107.02586* (2021).



Usynin, Dmitrii, Daniel Rueckert, and Georgios Kaissis. "Beyond gradients: Exploiting adversarial priors in model inversion attacks." *arXiv preprint arXiv:2203.00481* (2022).

Concrete mitigation: Differentially private stochastic gradient descent (DP-SGD)

The high-level algorithm:

1. Compute gradients for each individual sample (they represent independent clients)
2. Clip the calculated gradients to obtain a known sensitivity
3. Add the noise scaled by the sensitivity from step 2
4. Perform the gradient descent step

Abadi, Martin, et al. "Deep learning with differential privacy." *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016.

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\sum_i \left(\bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

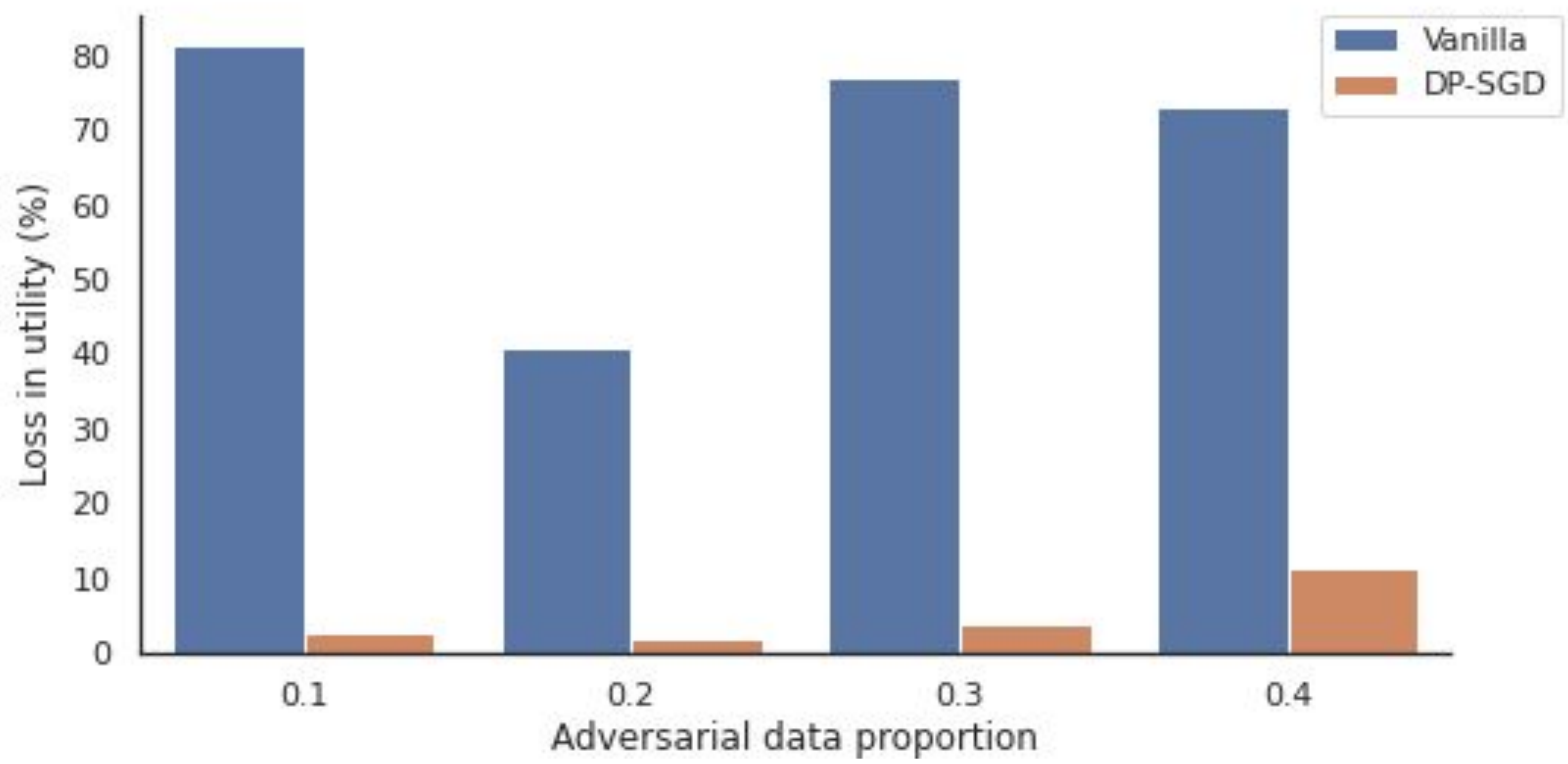
Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

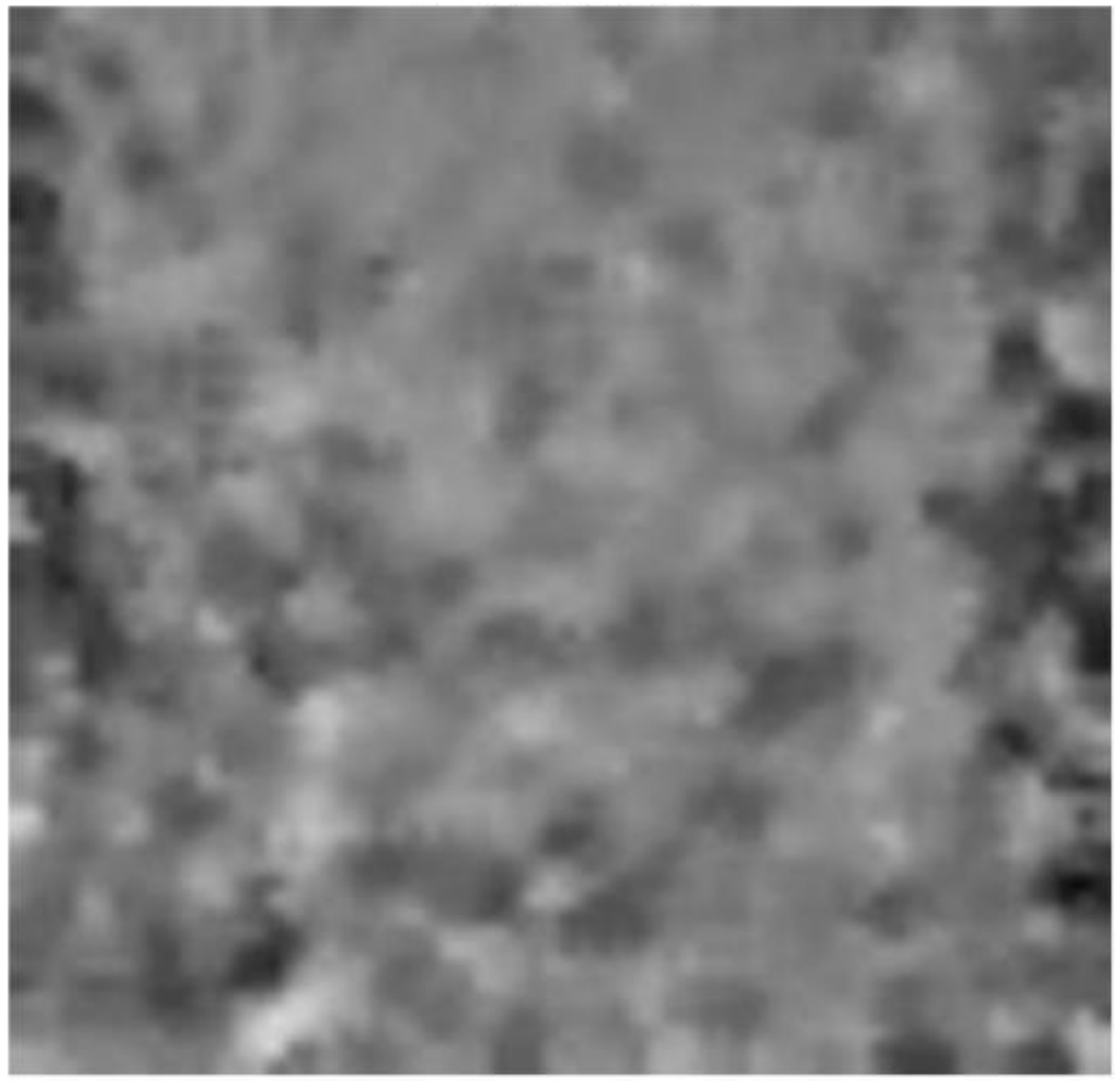
Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

Protecting against different attacks simultaneously:

Model poisoning



Model inversion



Usynin, Dmitrii, et al. "Can collaborative learning be private, robust and scalable?." *International Workshop on Distributed, Collaborative, and Federated Learning, Workshop on Affordable Healthcare and AI for Resource Diverse Global Health*. Springer, Cham, 2022.

There are many more question unanswered!

- How do defences interact with each other? *E.g. can you analyse encrypted updates?*
- What privacy parameters need to be selected to be defended against a certain attack type?
- How much utility impact must the data owner take to ensure that they are secure?
- Can we make models which are protected by-design?

Some common discussion points

Attacks can often represent the worst-case scenarios that are associated with the behaviour of ML models and can become more/less powerful at scale:

- The **more clients** you have, the **easier** it is to **conceal poisoning** attacks
- The **more devices** you have, the more **difficult** it is to perform **inversion**
- The **more clients** you have, the more personalised data they are likely to own -> **easier inference**
- **DP** can be **used at scale**, but it can result in **utility reduction** for **stricter privacy** settings (and is a nightmare to personalise)
- In general, **inversion attacks are oversold** and need many assumptions to hold in order to work
- **Inference** attacks are **possible at scale**, but come with a number of challenges
- Only **small number of datapoints** is required to **backdoor** a model (2% can be enough)

Any thoughts, comments or proposals?

Email me at du216@ic.ac.uk

I am also often active on privacy community Slacks e.g. OpenMined, prisec-ml etc.