Beyond Membership: Limitations of Add/Remove Adjacency in Differential Privacy

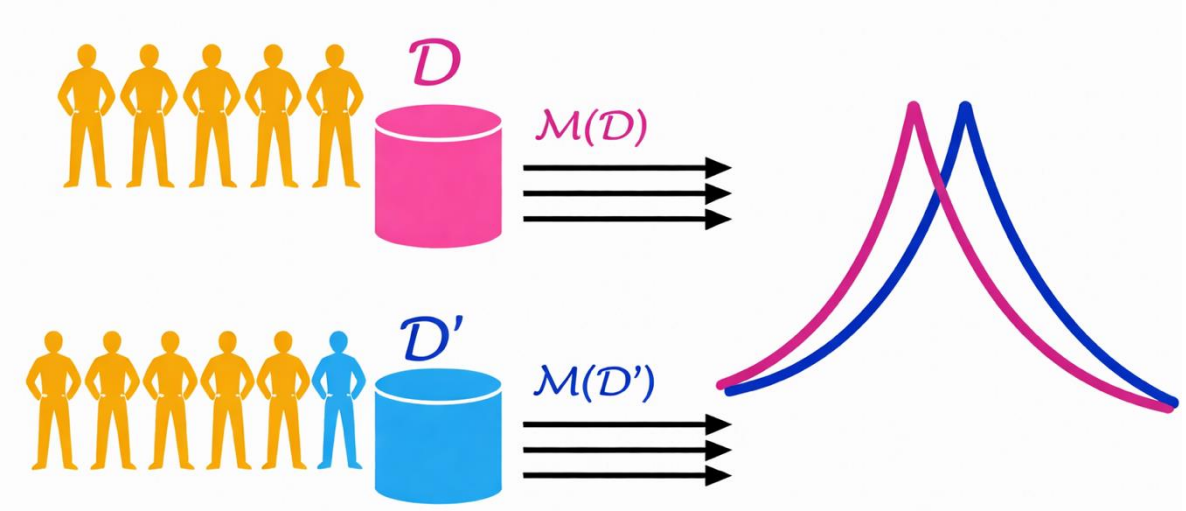
Gauri Pradhan[1], Joonas Jälkö[1], Santiago Zanella-Béguelin[2], Antti Honkela[1]
[1]University of Helsinki, Finland, [2]Microsoft, Cambridge, UK

UNIVERSITY OF HELSINKI

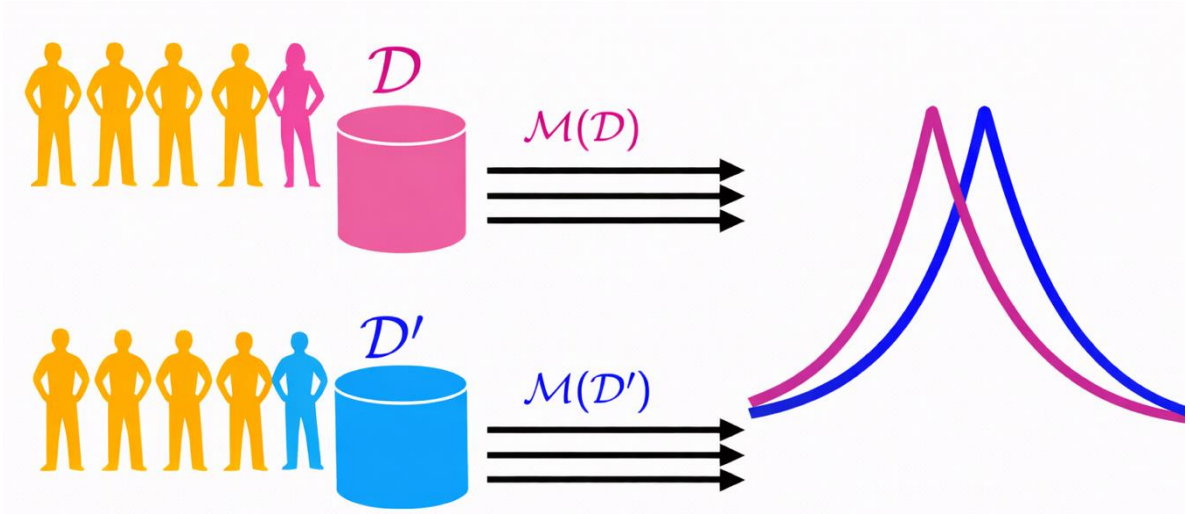
Differential Privacy (DP)

- A randomized algorithm $\mathcal{M}$ is $(\epsilon, \delta, \sim)$-differentially private if for all pairs of adjacent datasets, $\mathcal{D} \sim \mathcal{D}'$, and for all events S,

$$\Pr[\mathcal{M}(\mathcal{D}) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta.$$



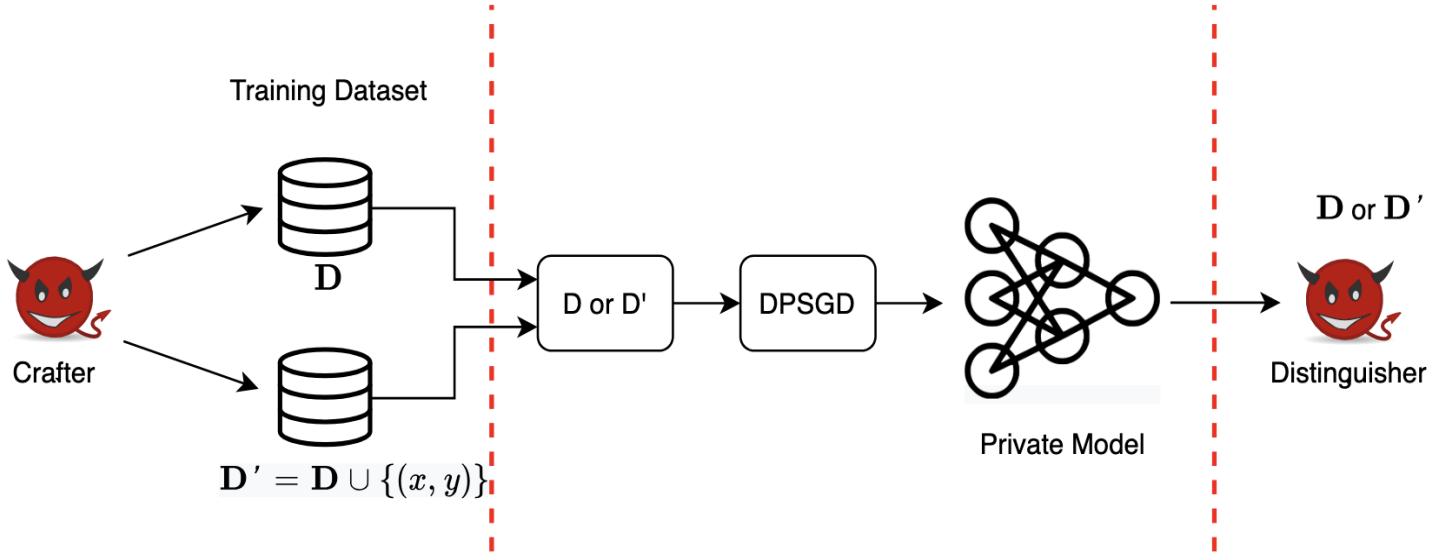
Add/Remove Adjacency
$(\mathcal{D} \sim_{AR} \mathcal{D}')$



Substitute Adjacency
$(\mathcal{D} \sim_{S} \mathcal{D}')$

Auditing DP

- Allows us to detect empirical privacy leakage by probing a target algorithm.
- Uses worst-case samples (*canaries*) to audit privacy guarantees.



Auditing setup for ML models trained with DP algorithms (such as DPSGD) from Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning (Nasr et al., 2021).

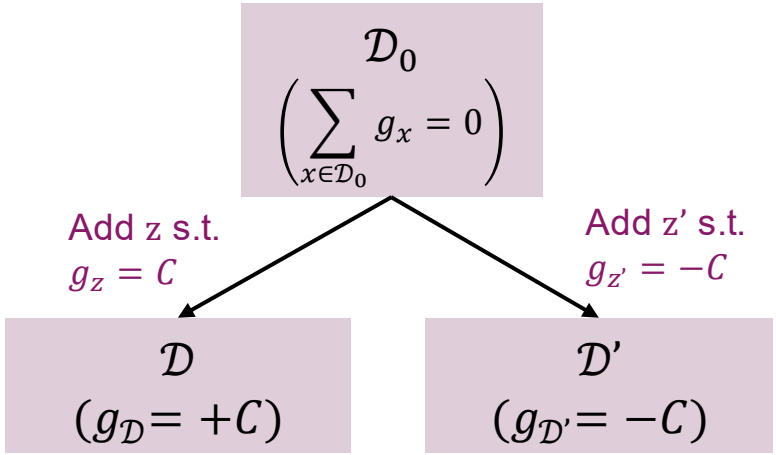
Auditing DP Under Substitute Adjacency

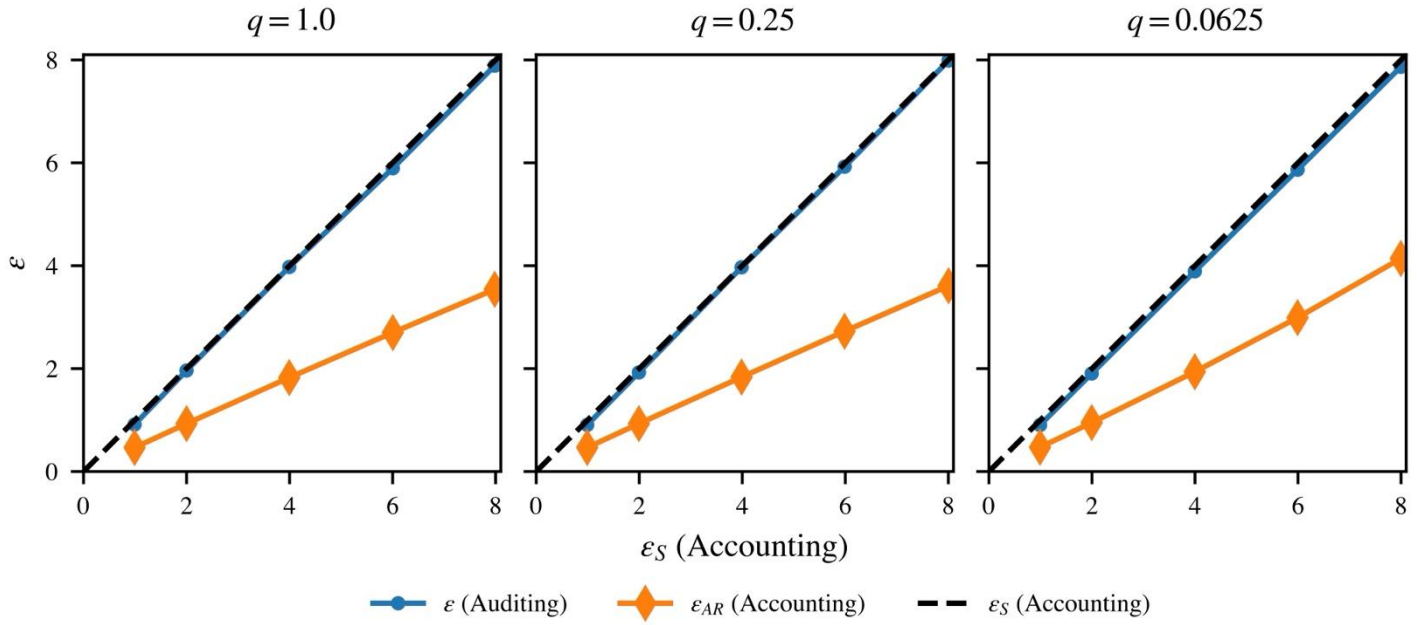
- We propose canaries for auditing DP under substitute adjacency.
- Using them, we show privacy leakage can **exceed** add/remove accounting guarantees.

Crafting Space	Type of Canary
Gradient	Crafted Dataset Canary
	Crafted Gradient Canary
Input	Crafted Input Canary
	Crafted Mislabeled Canary
	Adversarial Natural Canary

Crafting Dataset Canaries in Gradient Space

Crafting Space	Type of Canary
Gradient	**Crafted Dataset Canary**
	Crafted Gradient Canary
Input	Crafted Input Canary
	Crafted Mislabeled Canary
	Adversarial Natural Canary





Audits using worst-case dataset pairs yields tightest auditing results, showing violation of ε_{AR}(Accounting).

Canaries For Auditing Natural Datasets: Crafted Gradient Canary

Crafting Space	Type of Canary
Gradient	Crafted Dataset Canary
	Crafted Gradient Canary
Input	Crafted Input Canary
	Crafted Mislabeled Canary
	Adversarial Natural Canary

- Choose the least-updated parameter of the model (j^*),
- Form gradient canaries as,

$$g_z = (0,0,\dots,C,\dots,0),$$
$$g_{z'} = (0,0,\dots,-C,\dots,0)$$

- During training,

$$g_{\mathcal{D}} = g_{\mathcal{D}_0} + g_z$$
$$g_{\mathcal{D}'} = g_{\mathcal{D}_0} + g_{z'}$$

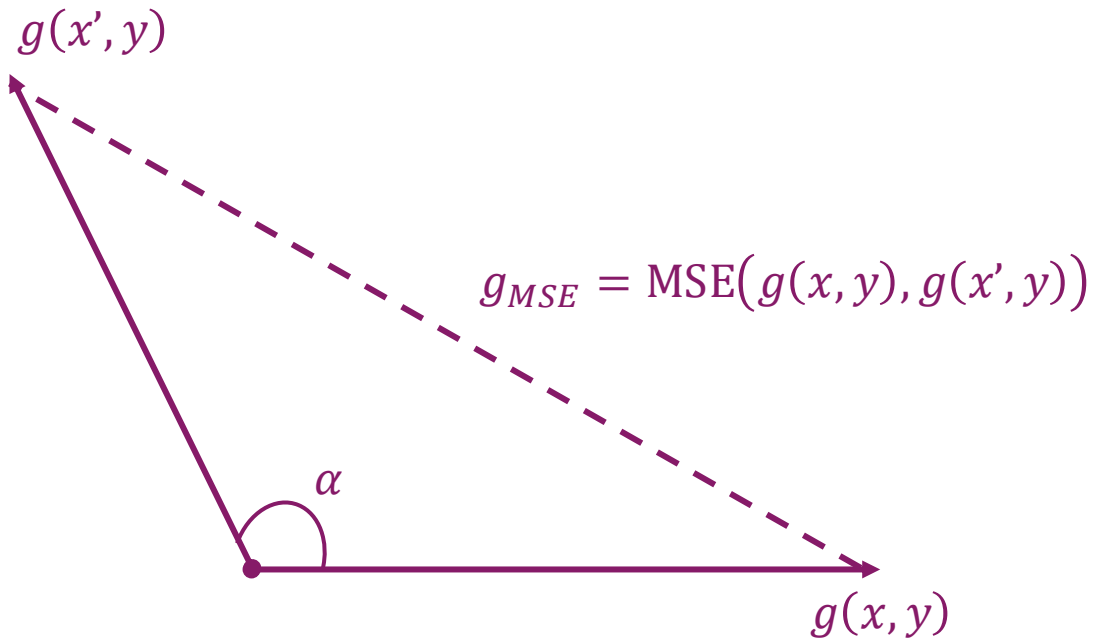
- Audit parameter drift per step (t),

$$\theta_t^{j^*} - \theta_0^{j^*}$$

Canaries For Auditing Natural Datasets: Crafted Input Canary

Crafting Space	Type of Canary
Gradient	Crafted Dataset Canary
	Crafted Gradient Canary
Input	**Crafted Input Canary**
	Crafted Mislabeled Canary
	Adversarial Natural Canary

- **Target** $(z \sim (x, y))$: least-confident sample under the reference model.
- **Canary** $(z' \sim (x', y))$: Obtained via crafting algorithm.
- **Audit Score**: $\text{logit}(z; \theta_t) - \text{logit}(z'; \theta_t)$

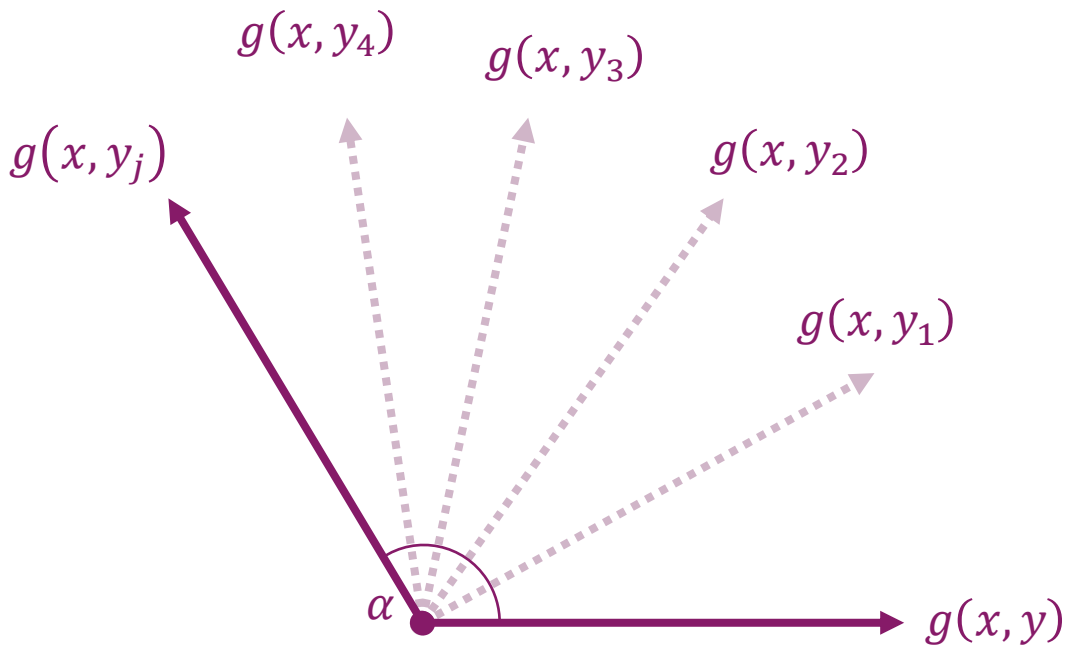


z' obtained by optimizing $\min_{(x',y)} g_{MSE} + \cos(\alpha)$

Canaries For Auditing Natural Datasets: Crafted Mislabeled Canary

Crafting Space	Type of Canary
Gradient	Crafted Dataset Canary
	Crafted Gradient Canary
Input	Crafted Input Canary
	Crafted Mislabeled Canary
	Adversarial Natural Canary

- **Target** $(z \sim (x, y))$: least-confident sample under the reference model.
- **Canary** $(z' \sim (x, y'))$: Obtained via crafting algorithm.
- **Audit Score**: $\text{logit}(z; \theta_t) - \text{logit}(z'; \theta_t)$

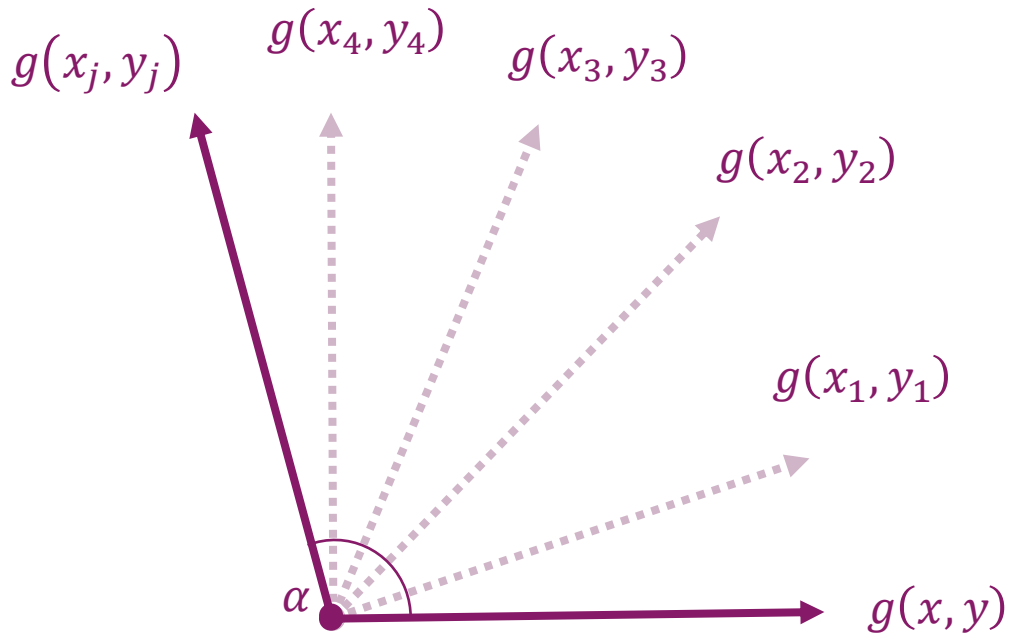


Find $z' \sim (x, y')$ by $\min_{(x, y_j)} \cos(\alpha)) \ \forall (x, y_j) \in \mathcal{Y}$

Canaries For Auditing Natural Datasets: Adversarial Natural Canary

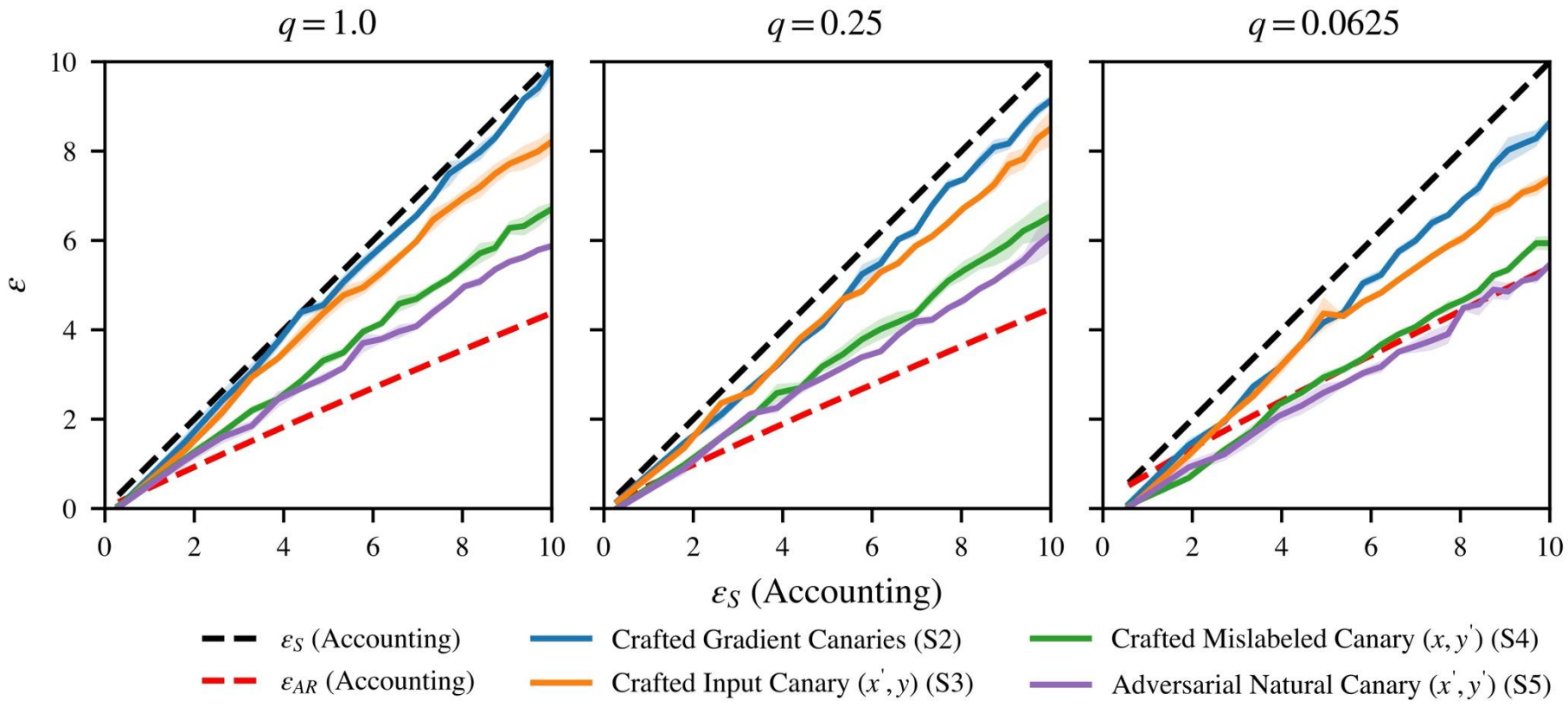
Crafting Space	Type of Canary
Gradient	Crafted Dataset Canary
	Crafted Gradient Canary
Input	Crafted Input Canary
	Crafted Mislabeled Canary
	Adversarial Natural Canary

- **Target** $(z \sim (x, y))$: least-confident sample under the reference model.
- **Canary** $(z' \sim (x', y'))$: Obtained via crafting algorithm.
- **Audit Score**: $\text{logit}(z; \theta_t) - \text{logit}(z'; \theta_t)$



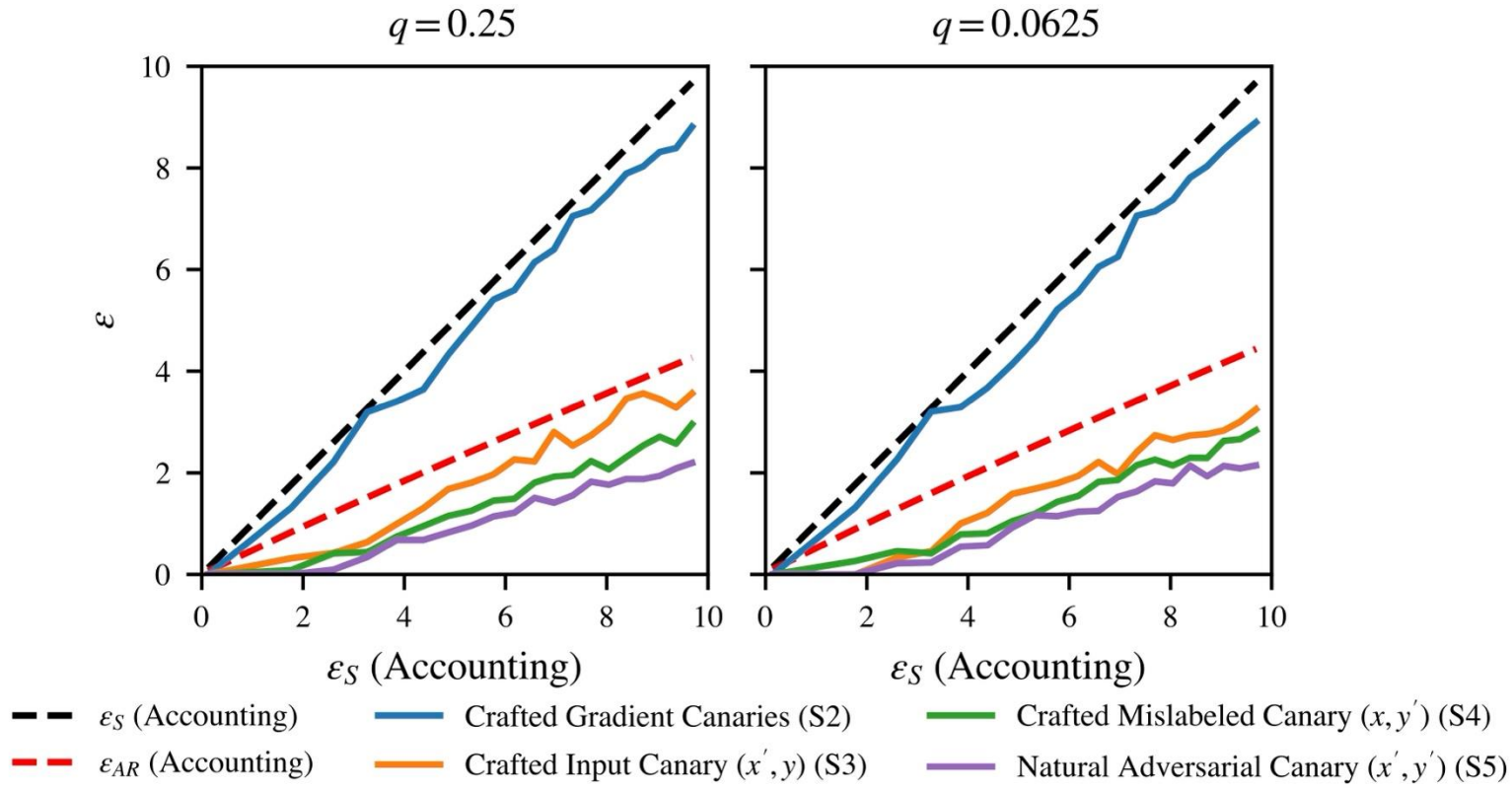
Find $z' \sim (x', y')$ by $\min_{(x_j, y_j)} \cos(\alpha))\ \forall (x_j, y_j) \in \mathcal{D}_{aux}$

Results: Auditing ViT-B-16 Model Fine-Tuned W/ CIFAR10



Running audits on pretrained ViT-B-16 model with final layer fine-tuned on CIFAR10 dataset reveal violations of ε_{AR} at high subsampling rate (q).

Results: Auditing MLP Trained From Scratch W/ Purchase100



Auditing MLP trained from scratch w/ Purchase100 using DP-Adam, input-space canaries lose efficacy, but gradient canaries still yield violations of ε_{AR}.

Conclusion

- **Add/remove DP can overestimate protection** against substitute-style privacy attacks.
  - Canary-based auditing reveals this gap,
  - Gradient-space canaries are particularly effective.
- Violations are **stronger** for,
  - higher subsampling rates (q),
  - for supervised fine-tuning tasks.