UBA-Inf: Unlearning Activated Backdoor Attack with Influence-Driven Camouflage

Zirui Huang[1], Yunlong Mao[1], Sheng Zhong[1]

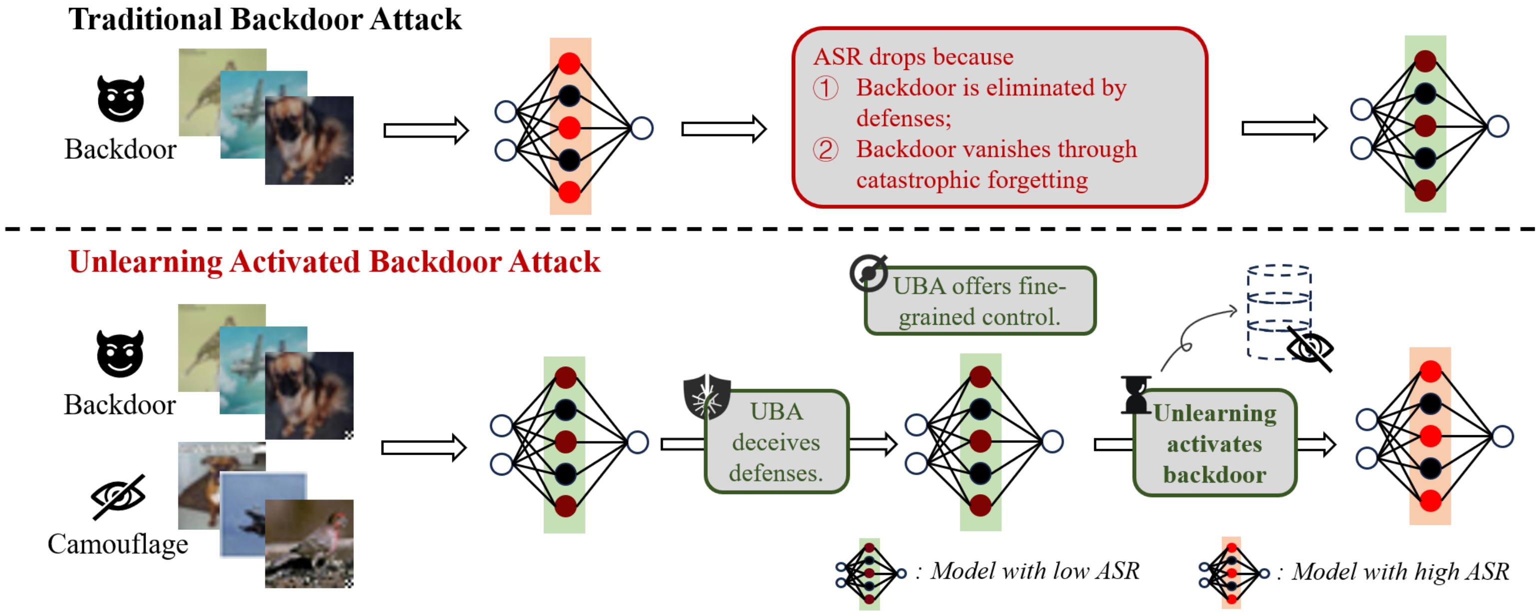
[1]*State Key Laboratory for Novel Software Technology, Nanjing University, China*

huangzirui@smail.nju.edu.cn, *{maoyl, zhongsheng}@nju.edu.cn*

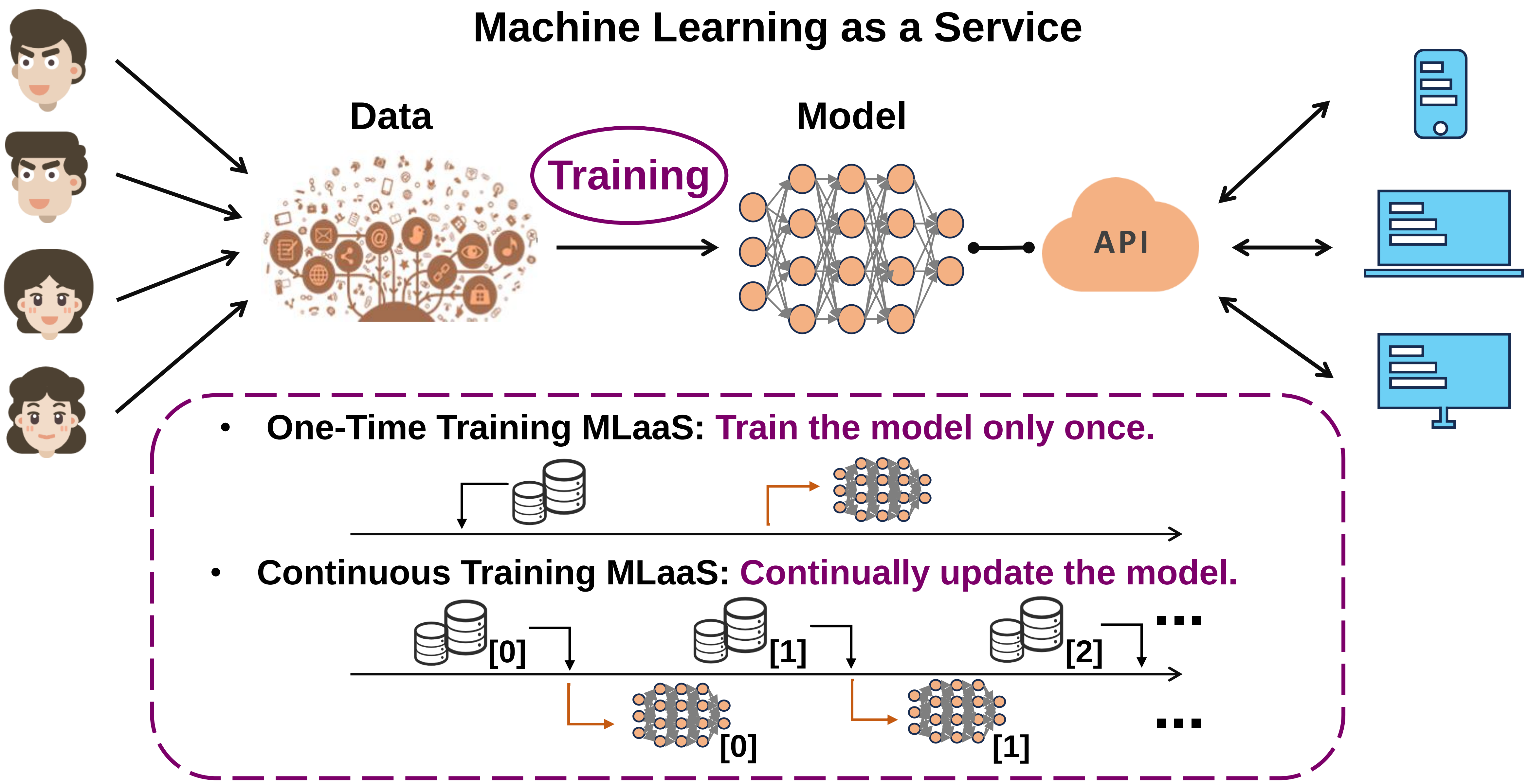
The 33rd **USENIX Security** Symposium

Let’s begin with some easy take-aways

- *Uncovering vulnerabilities in machine unlearning;*
- *Combining backdoor attacks and unlearning;*
- *Advancing persistent backdoor attacks in continual leaning.*



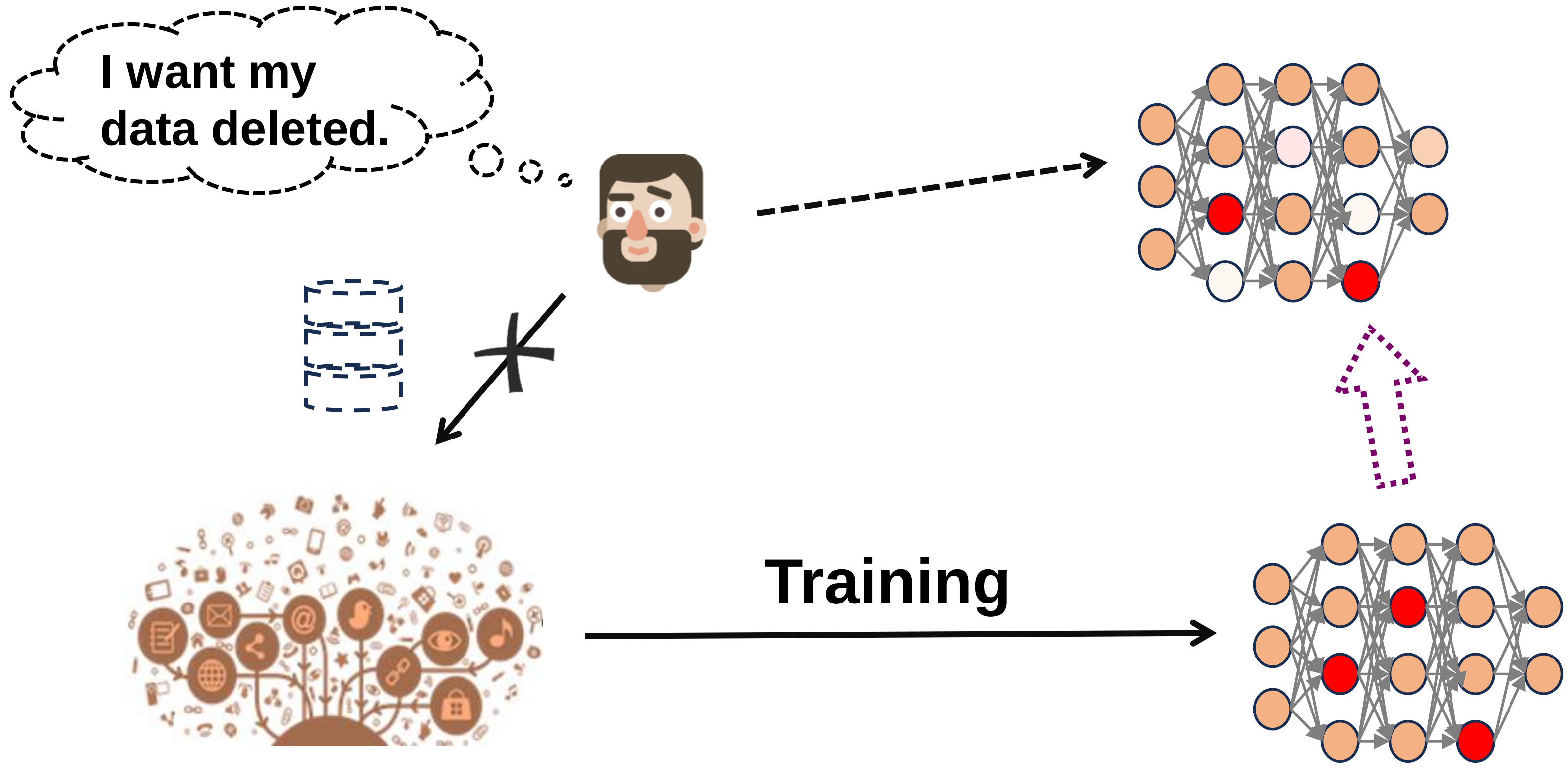
Background: MLaaS (One-Time & Continual Training)



Background: Machine unlearning

Machine Unlearning

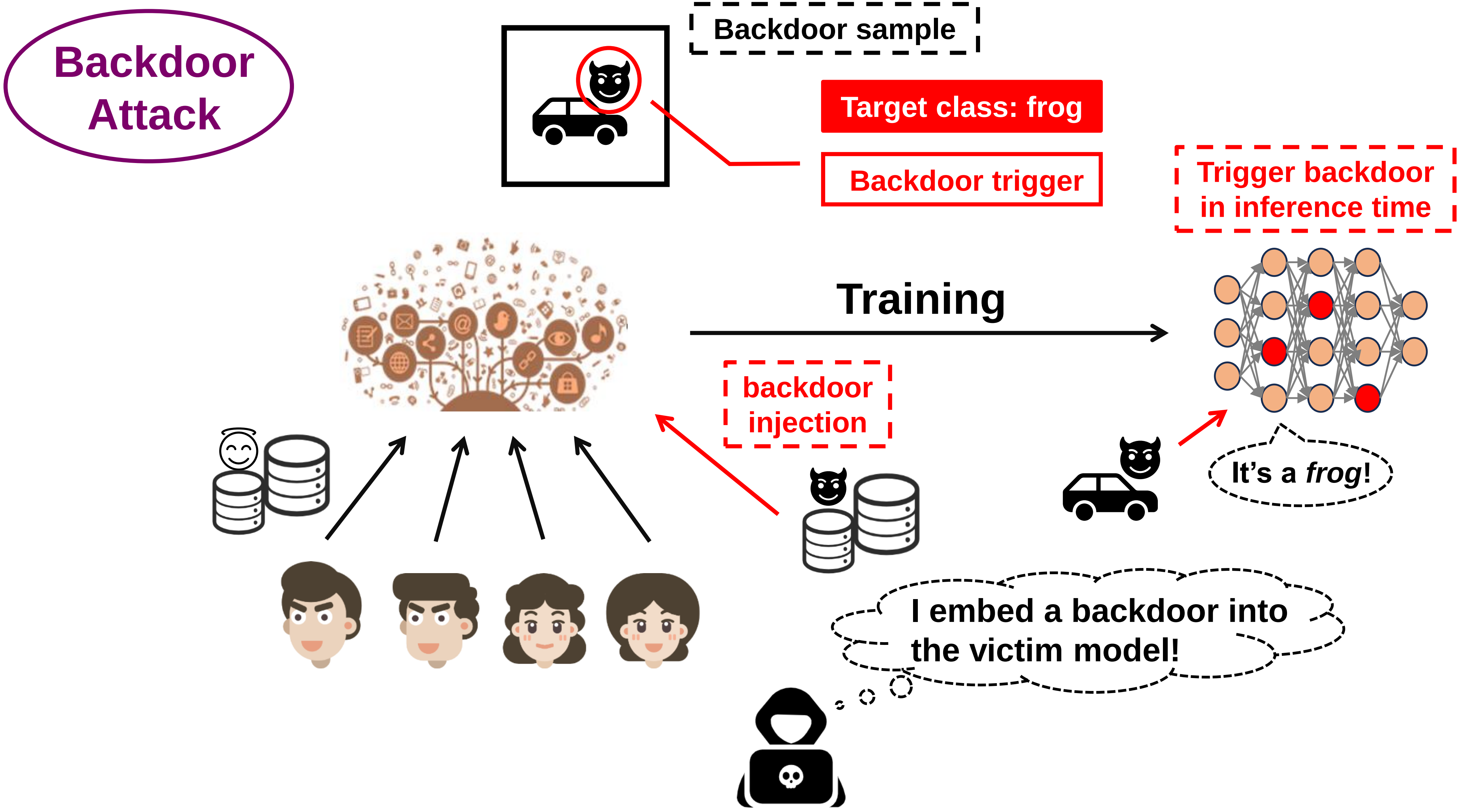
Goal of unlearning:
- The model after unlearning should be as if that data had never been part of the training process



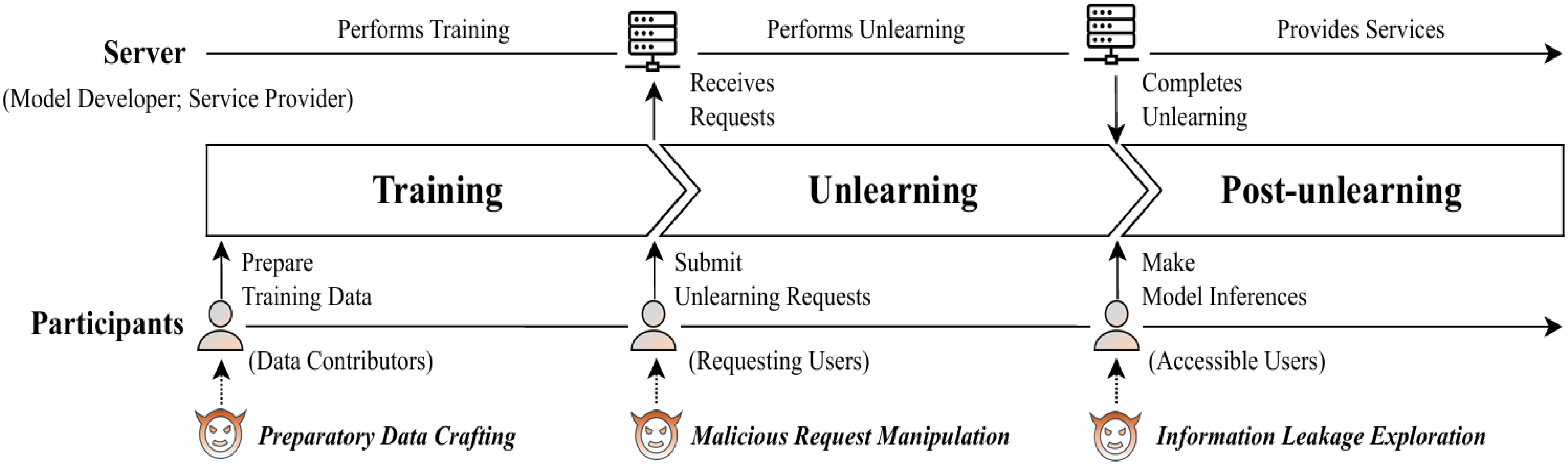
Motivations for unlearning
- **Access revocation** (think unlearning private and copyrighted data).
- **Model correction & editing** (think toxicity, bias, stale/dangerous knowledge removal).

Approaches to unlearning:
- **Exact unlearning** (retraining-based)
- **Approximate unlearning** (directly modify model parameters)

Background: Machine unlearning & Backdoor attack



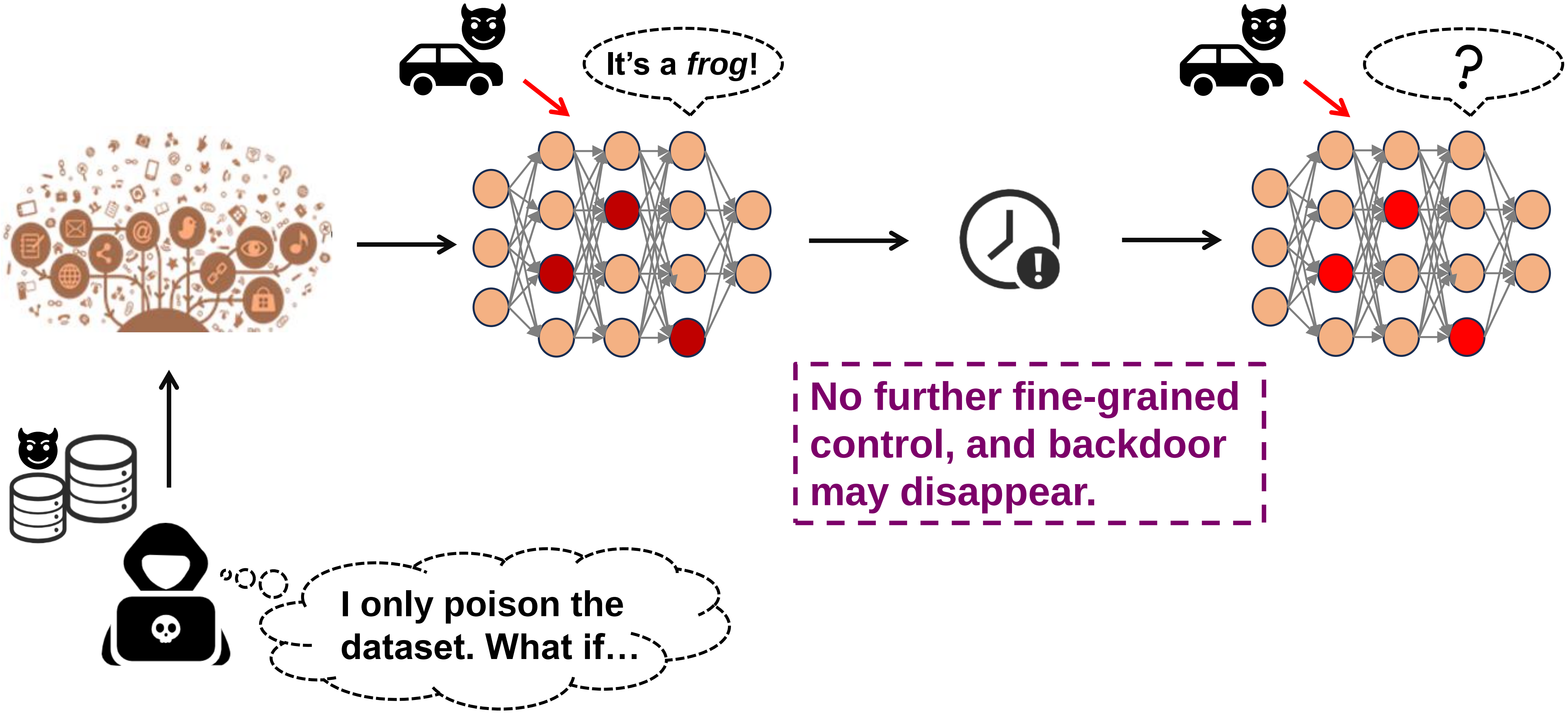
Motivation: There exist various unlearning vulnerabilities.



Machine unlearning is vulnerable!

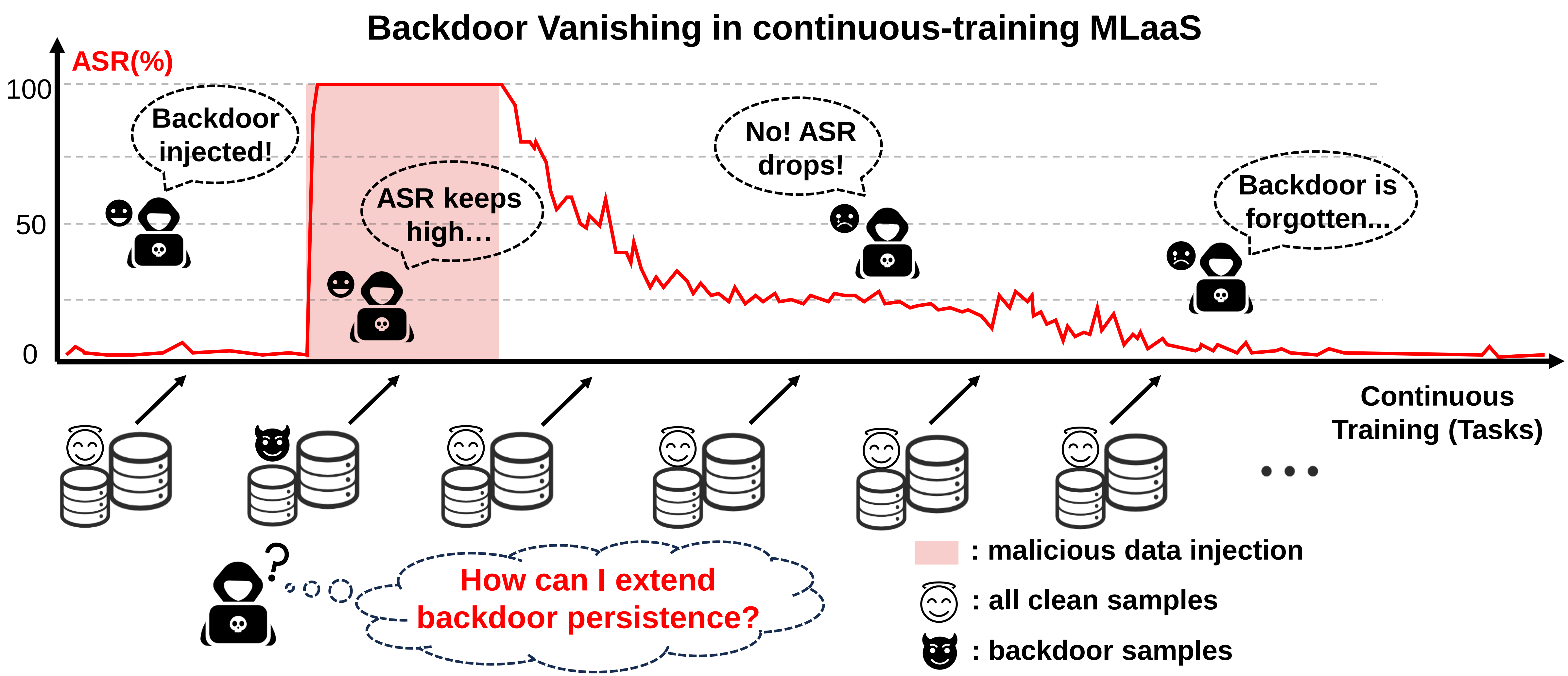
Reference: Liu Z, Ye H, Chen C, et al. Threats, attacks, and defenses in machine unlearning: A survey[J]. arXiv preprint arXiv:2403.13682, 2024.

Motivation: Traditional backdoor lacks fine-grained control.

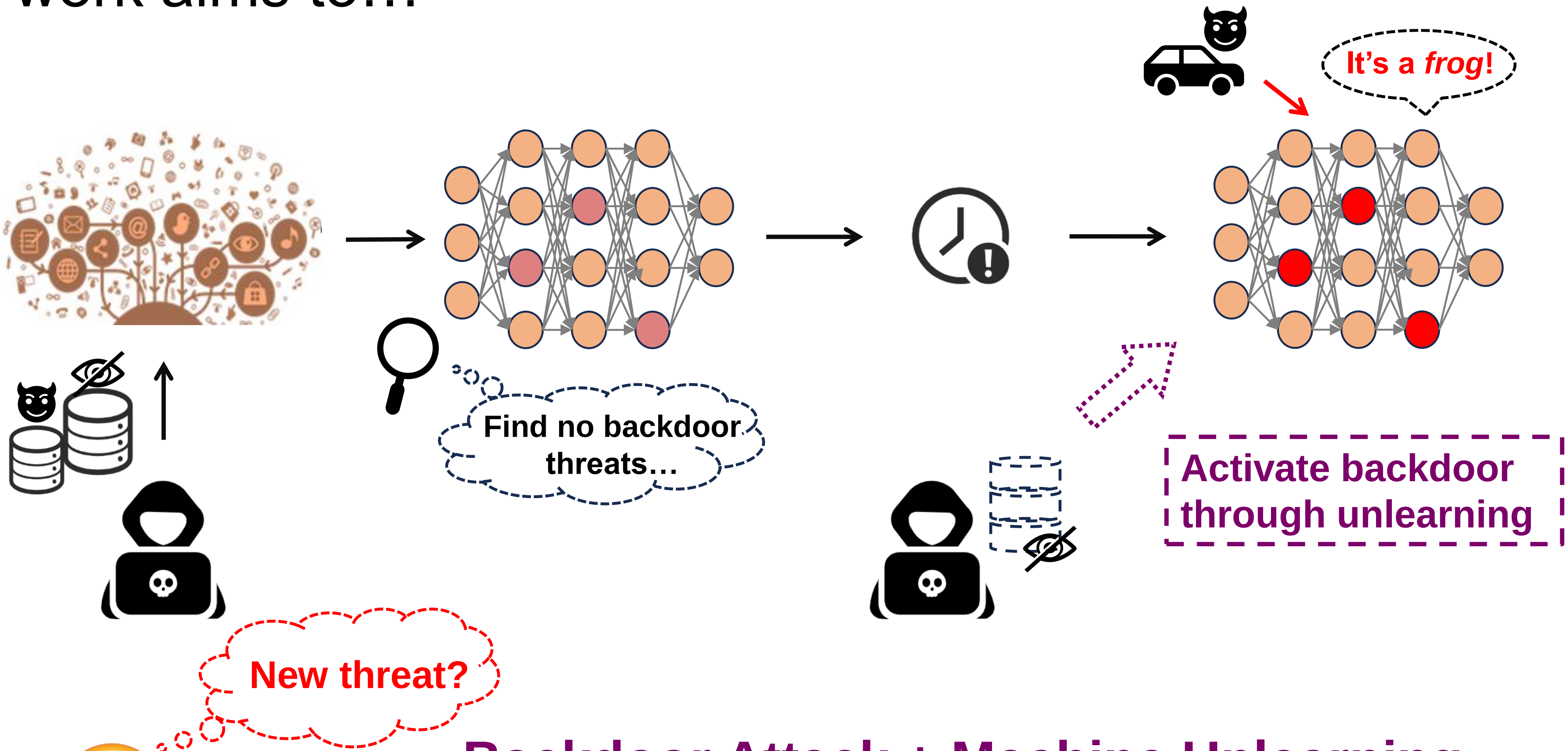


No further fine-grained control, and backdoor may disappear.

Motivation: Backdoor vanishes in continuous training.



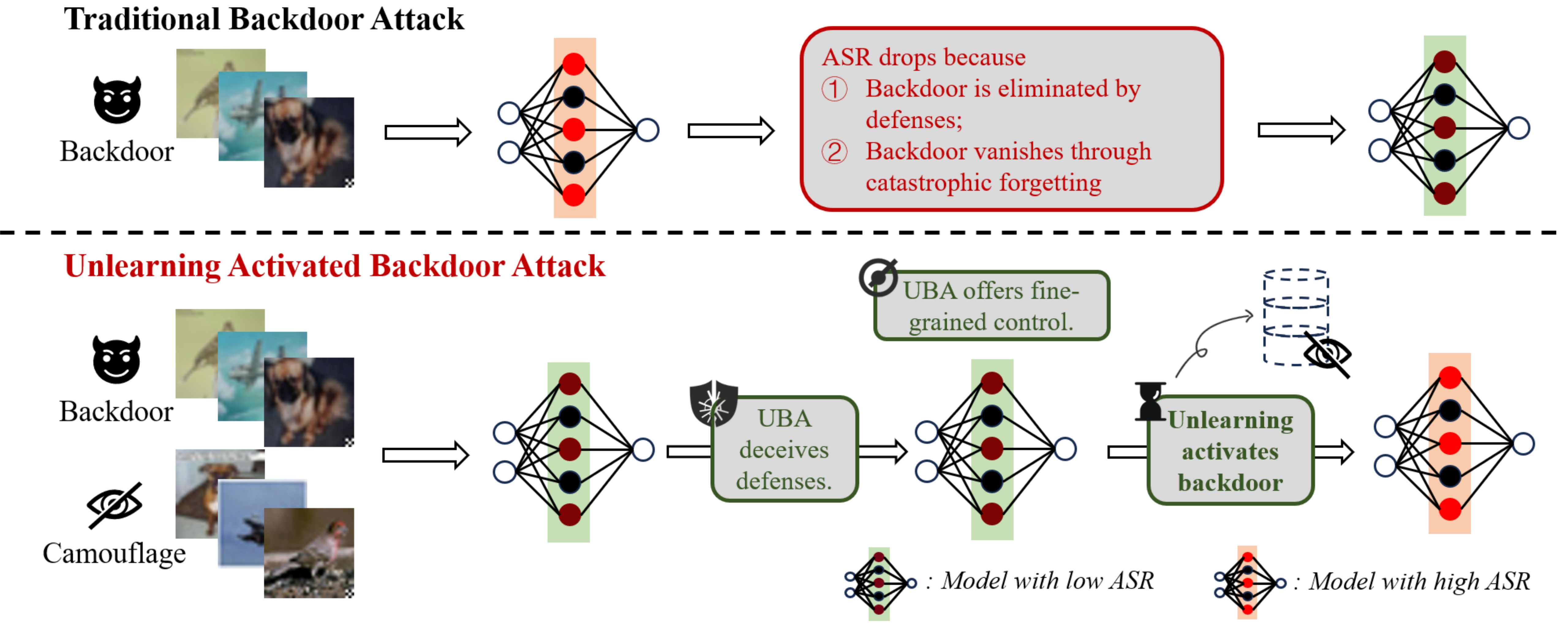
Our work aims to…



Backdoor Attack + Machine Unlearning

Method: Unlearning-activated Backdoor Attack

UBA-Inf



Threat model

Adversary:

- ☐ The ability to add and delete data points from target model with requests.
- ☐ An auxiliary dataset $\boldsymbol{D_{atk}}$
- ☐ A surrogate model $\boldsymbol{\theta_s}$ trained on public dataset.
- ☐ A prepared backdoor generation algorithm $\boldsymbol{B}(\cdot)$

***Goal**: **high** Benign Accuracy (**BA**) and high Attack Success Rate (**ASR**) when triggering backdoor*

Service Provider:

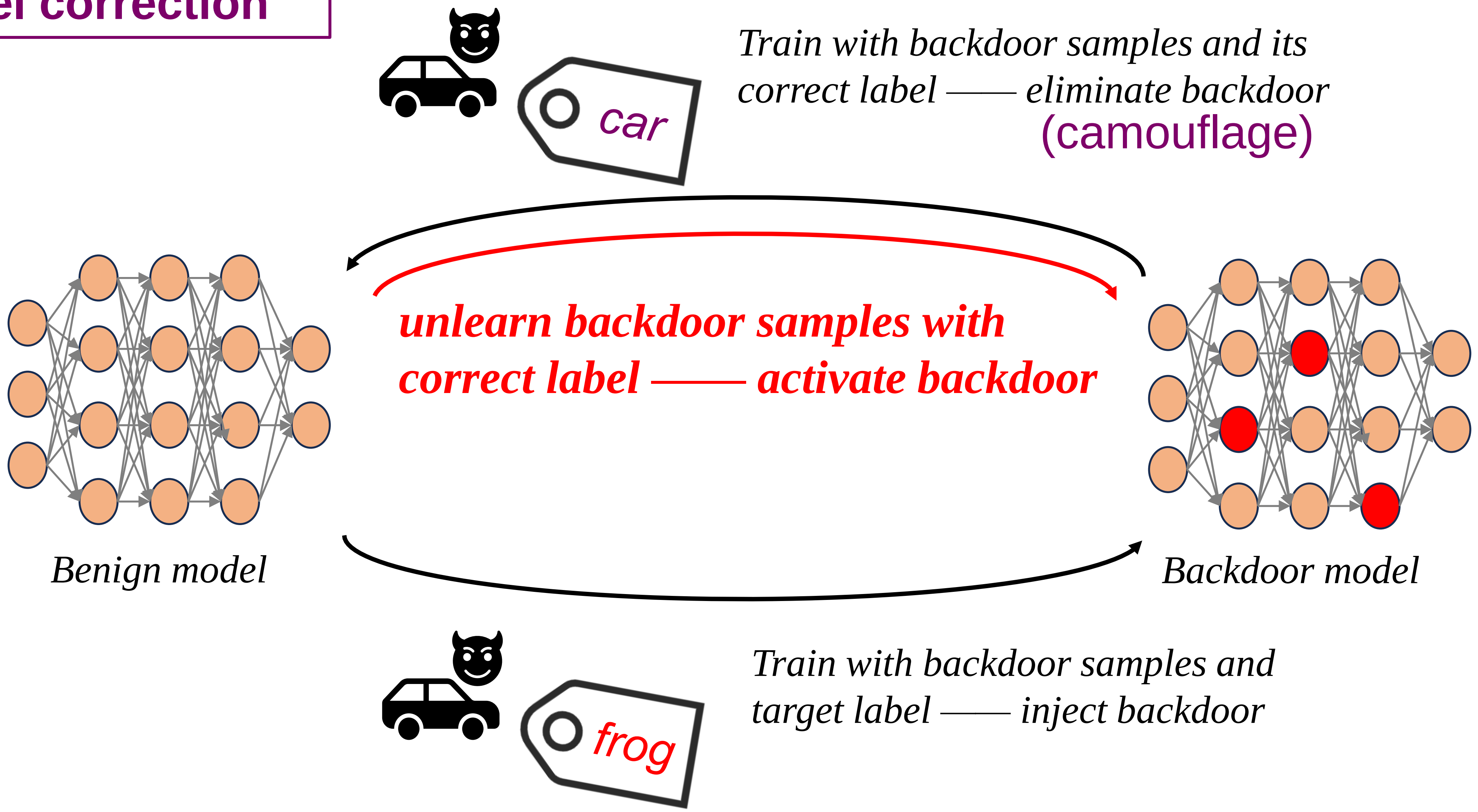
- ☐ Collect data and train the target model.
- ☐ Unlearning sensitive samples as requested.
- ☐ Perform defenses against potential attacks.

Key to design:

1. **How to construct effective camouflage samples?**
2. **How to implement the whole attack pipeline?**

Method: UBA-Inf design rationale

Label correction



Method: UBA-Inf design rationale

Influence function

In practice, it's not adequately effective to merely correct the label of backdoor samples…

State	Method	CIFAR-10		MNIST		GTSRB		Tiny	
		BA(%)	ASR(%)	BA(%)	ASR(%)	BA(%)	ASR(%)	BA(%)	ASR(%)
before unlearn	UBA-Inf	93.26	**21.94**	99.50	**29.42**	98.34	**22.15**	55.56	**16.57**
	BAMU	93.19	36.71	99.47	90.14†	98.51	28.44	56.20	37.95
after full retrain	UBA-Inf	93.34	**100.00**	99.64	**100.00**	97.85	**99.89**	56.09	**92.26**
	BAMU	93.12	100.00	99.58	100.00†	98.23	99.63	55.90	88.73
after PUMA	UBA-Inf	89.50	**80.44**	98.27	**81.51**	98.27	**81.51**	50.06	**71.72**
	BAMU	89.97	50.10	98.39	99.93†	94.90	64.13	50.02	56.21
after GBU	UBA-Inf	90.53	**83.60**	98.28	**89.01**	95.18	**80.20**	49.98	**64.26**
	BAMU	90.11	52.53	98.47	92.49†	94.82	59.71	50.24	47.15

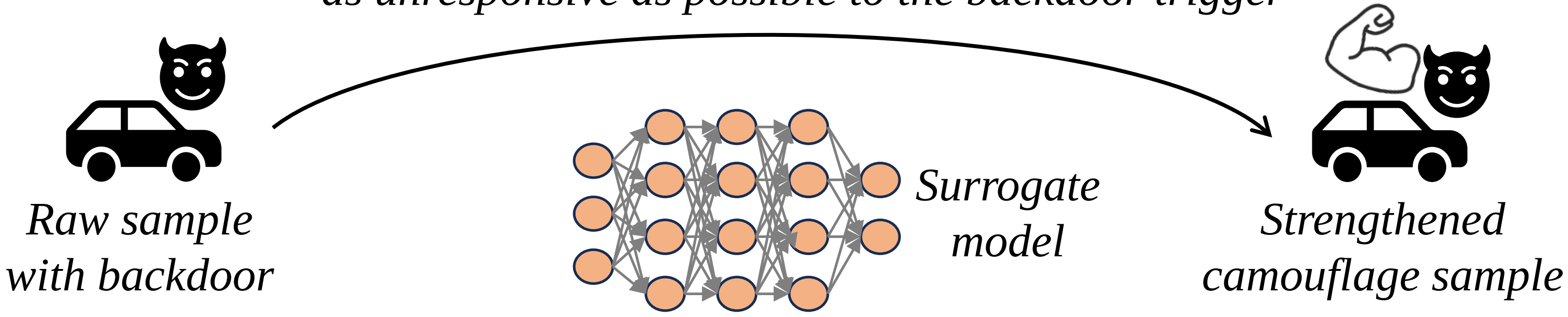
† BAMU fails in MNIST with ASR higher than 80%, which completely has no camouflage effect.

In some cases, the backdoor is not camouflaged…

In some cases, the backdoor is not effectively activated…

*Use **Influence function** to strengthen camouflage samples!*

- *Perturb through influence function to make the model as unresponsive as possible to the backdoor trigger*



Method: UBA-Inf camouflage

UBA-Inf Camouflage Generation Algorithm

Adversary Knowledge

- $\boldsymbol{\theta_s}$: surrogate model trained on public-out-of-distribution dataset
- $\boldsymbol{D_{atk}}$: auxiliary dataset in the same distribution of real dataset.
- $\boldsymbol{B}(\cdot)$: backdoor generation algorithm

Label Correction

- Backdoor samples $\boldsymbol{D_{bd}} = \{B\big((x,y)\big) | (x,y) \in D_{atk}\}$
- Label correction $\boldsymbol{D_{cm}} = \{(B_X(x), y) \mid (x,y) \in D_{atk} \wedge y \neq y_{tgt}\}$

Influence Function

- Analyze the direction of camouflage perturbation that makes the model as unresponsive as possible to the backdoor trigger

$$
\begin{aligned}
\mathcal{I}_{pert,loss}(\tilde{z}, D_{bd}) &= \mathop{\mathbf{E}}_{z' \in D_{bd}} (\mathcal{I}_{pert,loss}(\tilde{z}, z')) \\
&= - \mathop{\mathbf{E}}_{z' \in D_{bd}} (\nabla_\theta \ell(z', \theta^*_{s,i})^\top)(\frac{1}{m} \sum_{i=1}^{m} \nabla^2_\theta \ell(z_i, \theta^*_{s,i}))^{-1} \nabla_x \nabla_\theta \ell(\tilde{z}, \theta^*_{s,i}),
\end{aligned}
$$

Iterative Optimization

- Fine-tune $\boldsymbol{\theta_s}$, optimize D_{cm} through $\boldsymbol{I_{\{pert,loss\}}}$

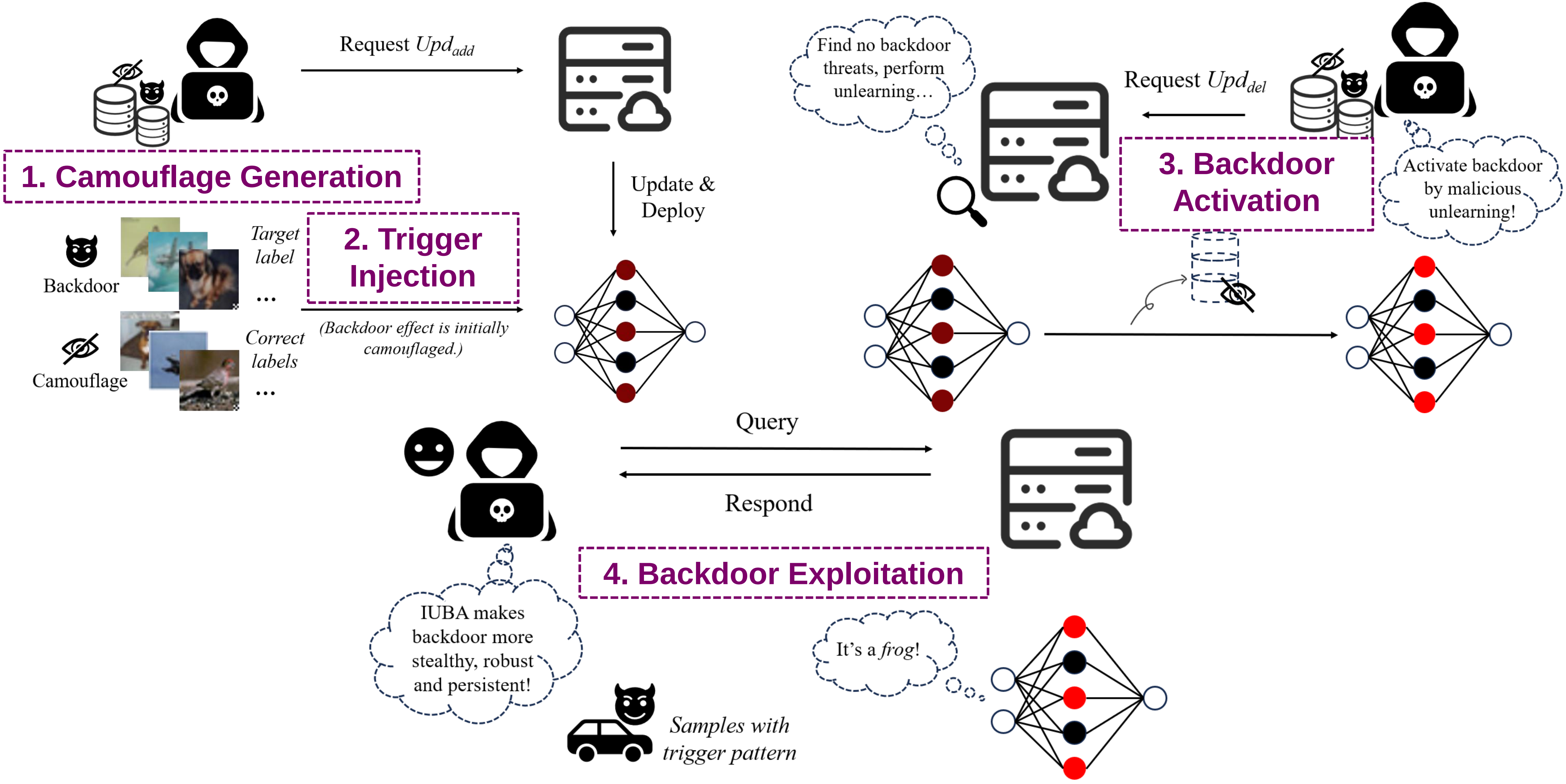
Algorithm 1 UBA-Inf Camouflage Generation Algorithm

Input: θ^*_s (pre-trained surrogate model)
D_{bd} (backdoor samples)
D_{atk} (auxiliary samples)
B_X, y_{tgt} (backdoor trigger and target class)
N (total iteration epochs)
n, ε, α (adversarial perturbation parameters)

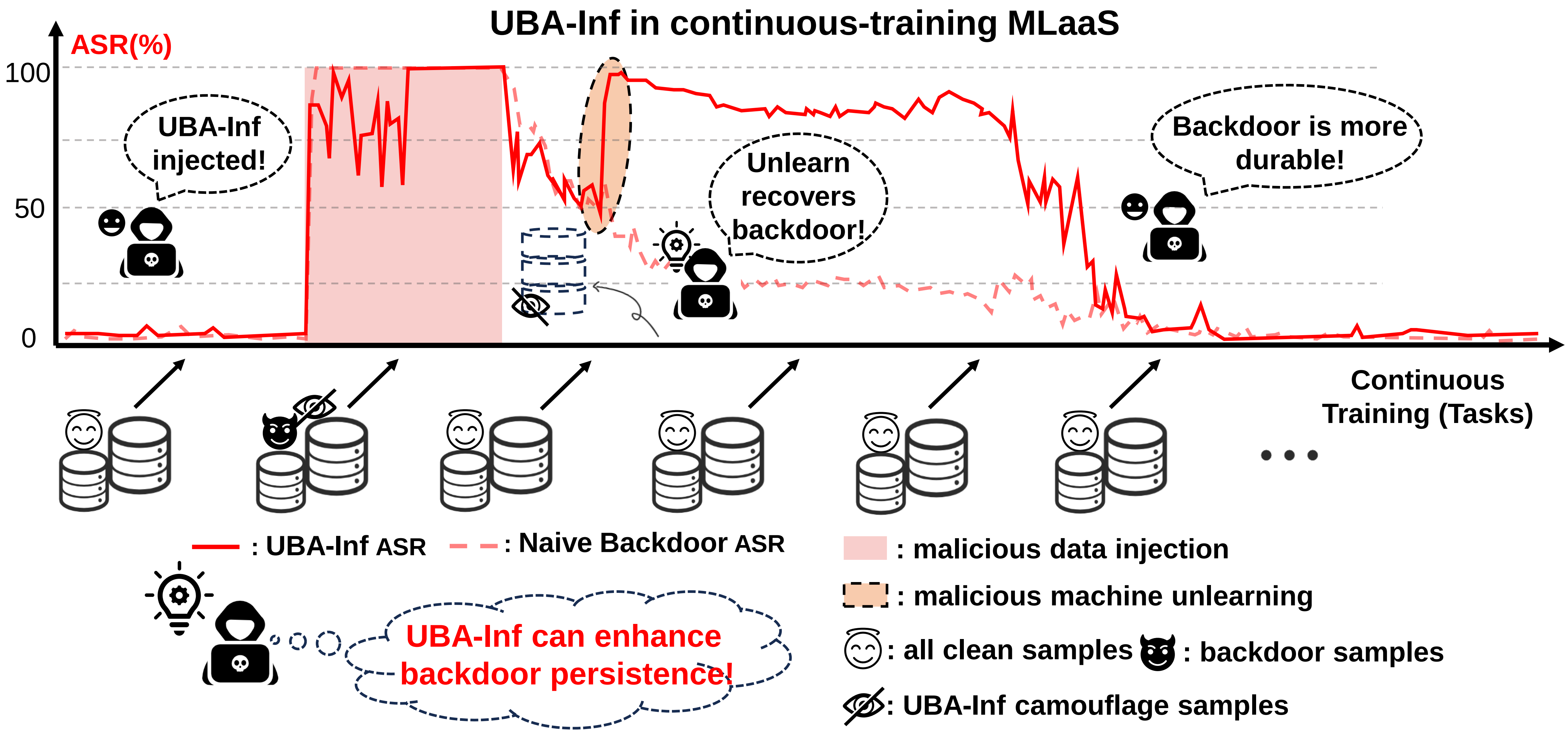
Output: D_{cm} (UBA-Inf camouflage samples)

$\theta^*_{s,0} \leftarrow finetune(\theta^*_s, D_{atk})$
$D_{cm,cl} \leftarrow \{ (x,y) | (x,y) \in D_{atk} \wedge y \neq y_{tgt} \}$
$D_{cm,0} \leftarrow \{ (B_X(x), y) | (x,y) \in D_{cm,cl} \}$
$D_{atk,0} = (D_{atk} \setminus D_{cm,cl}) \cup D_{bd} \cup D_{cm,0}$
for each iteration $i \in [1, N]$ **do**
  $\theta^*_{s,i} \leftarrow finetune(\theta^*_{s,0}, D_{atk,i-1})$
  $D_{cm,i} \leftarrow \emptyset$
  for $\tilde{z} \in D_{cm,i-1}$ **do**
    $\tilde{z}^0 \leftarrow \tilde{z}$
    for each perturbation $j \in [1, n]$ **do**
      $I_{pert,loss}(\tilde{z}^{j-1}, D_{bd}) \leftarrow \mathop{\mathbf{E}}_{z' \in D_{bd}} (I_{pert,loss}(\tilde{z}^{j-1}, z'))$
      $\tilde{z}^j \leftarrow \Pi_{\varepsilon, \tilde{z}_0}(\tilde{z}^{j-1} + \alpha sign(I_{pert,loss}(\tilde{z}^{j-1}, D_{bd})))$
    end for
    $D_{cm,i} \leftarrow D_{cm,i} \cup \{\tilde{z}^n\}$
  end for
  $D_{atk,i} \leftarrow (D_{atk,i-1} \setminus D_{cm,i-1}) \cup D_{cm,i}$
end for
$D_{cm} \leftarrow D_{cm,N}$
return D_{cm}

Method: UBA-Inf implementation in One-time training MLaaS



Method: UBA-Inf implementation in Continuous Training MLaaS



Evaluation: Effectiveness

Camouflage effect of UBA-Inf achieves rather low ASR.

Activation effect of UBA-Inf achieves high ASR close to 100%.

Shards		BadNets[1] BA(%)	BadNets[1] ASR(%)	Blended[2] BA(%)	Blended[2] ASR(%)	LC[3] BA(%)	LC[3] ASR(%)	Sig[4] BA(%)	Sig[4] ASR(%)
CIFAR-10									
shard=3	conceal	90.76	**12.26**	90.62	**22.72**	90.43	**23.54**	90.96	**9.24**
	unlearn	90.65	***99.98***	90.26	***89.92***	90.30	***88.65***	90.95	***89.42***
shard=5	conceal	88.74	**17.01**	88.30	**22.88**	88.62	**27.12**	88.82	**17.50**
	unlearn	88.68	***99.94***	88.59	***91.82***	88.11	***88.00***	88.66	***96.36***
MNIST									
shard=3	conceal	99.58	**6.58**	99.70	**25.03**	99.66	**0.28**	99.63	**0.38**
	unlearn	99.66	***100.00***	99.66	***100.00***	99.65	***73.50***	99.68	***65.35***
shard=5	conceal	99.64	**1.90**	99.67	**18.33**	99.56	**0.35**	99.56	**0.48**
	unlearn	98.57	***100.00***	99.67	***100.00***	99.53	***54.03***†	99.49	***34.66***†
GTSRB									
shard=3	conceal	99.59	**23.31**	98.36	**24.32**	98.23	**0.03**	98.32	**5.48**
	unlearn	99.61	***100.00***	98.50	***88.86***	98.24	***4.61***†	98.13	***72.30***
shard=5	conceal	99.59	**15.21**	97.98	**24.60**	98.27	**0.03**	98.01	**10.01**
	unlearn	99.58	***100.00***	97.96	***83.24***	97.41	***3.15***†	97.76	***69.58***
Tiny									
shard=3	conceal	51.47	**20.60**	51.38	**20.12**	52.03	**3.23**	51.81	**10.25**
	unlearn	51.40	***87.73***	52.15	***82.27***	51.45	***47.35***†	51.73	***79.66***
shard=5	conceal	48.36	**24.60**	47.91	**16.46**	48.12	**5.83**	48.36	**9.35**
	unlearn	47.63	***82.47***	48.06	***85.21***	48.02	***32.75***†	47.45	***79.23***

† Similar to full retrain, LC does not work properly on GTSRB and Tiny, while Sig has problems with SISA on MNIST. To avoid such a situation, the UBA-Inf adversary can choose a proper backdoor attack alternatively.

*Backdoor effectiveness evaluation for **exact machine unlearning** SISA. Two different numbers of training data shards are considered.*

Table 5: Backdoor effectiveness evaluation for PUMA.

Dataset	Models	conceal BA(%)	conceal ASR(%)	unlearn BA (%)	unlearn ASR(%)
CIFAR-10	PARN-18	93.26	**21.94**	89.50	***80.44***
	ResNet-34	93.47	**22.10**	89.91	***80.60***
	VGG-16	90.71	**22.24**	89.52	***89.68***
MNIST	PARN-18	99.50	**29.42**	98.27	***81.51***
GTSRB	PARN-18	98.34	**22.15**	98.19	***81.46***
Tiny	PARN-18	55.56	**16.57**	50.06	***71.72***

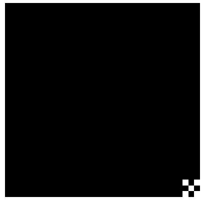
Table 6: Backdoor effectiveness evaluation for GBU.

Datasets	Models	conceal BA(%)	conceal ASR(%)	unlearn BA(%)	unlearn ASR(%)
CIFAR-10	PARN-18	93.26	**21.94**	90.53	***83.60***
	ResNet-34	93.47	**22.10**	90.19	***86.25***
	VGG-16	90.71	**22.24**	89.28	***89.96***
MNIST	PARN-18	99.50	**29.42**	98.28	***89.01***
GTSRB	PARN-18	98.34	**22.15**	95.18	***80.20***
Tiny	PARN-18	55.56	**16.57**	49.98	***64.26***

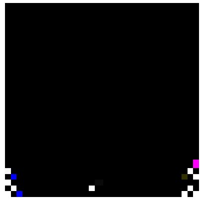
*Backdoor effectiveness evaluation for **approximate machine unlearning methods** like PUMA and GBU.*

Evaluation: Stealthiness before unlearning

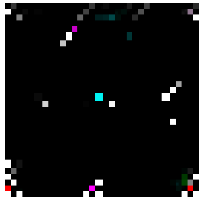
- UBA-Inf improves backdoor stealthiness. For example, for defenses that reverse the backdoor trigger, UBA-Inf can confuse the scanner so that the backdoor cannot be correctly revealed.



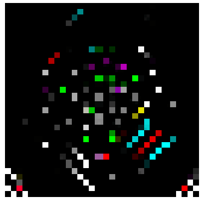
The real BadNet trigger (3 × 3, right-bottom)



Reversed trigger by NC without camouflage.



Reversed trigger by NC with BAMU camouflage.



Reversed trigger by NC with UBA-Inf camouflage.

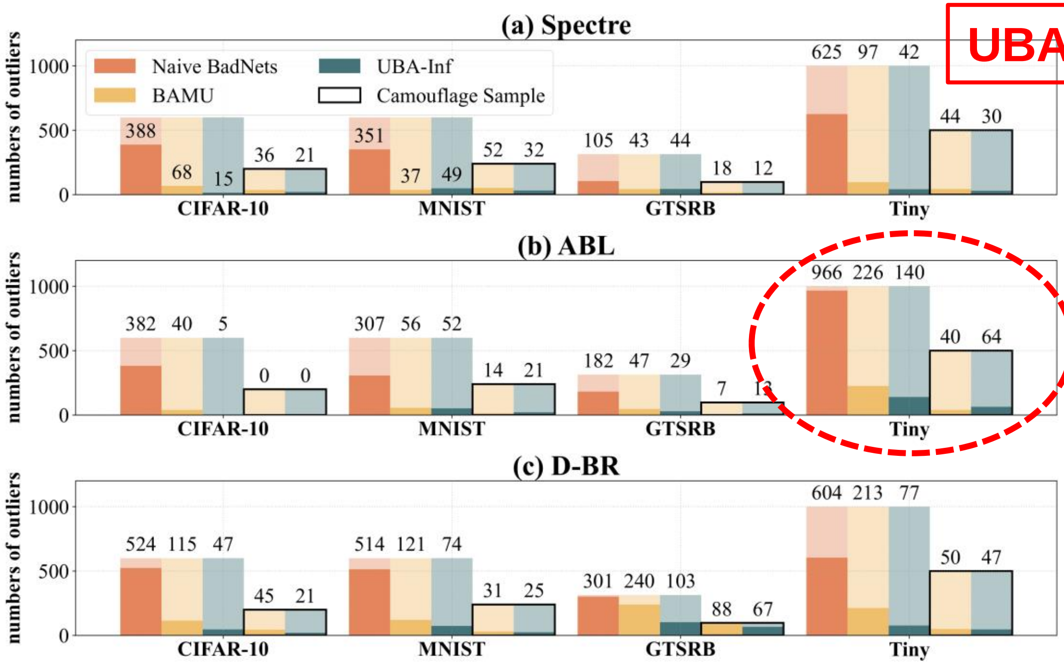
Raw backdoor can be easily reversed and revealed.

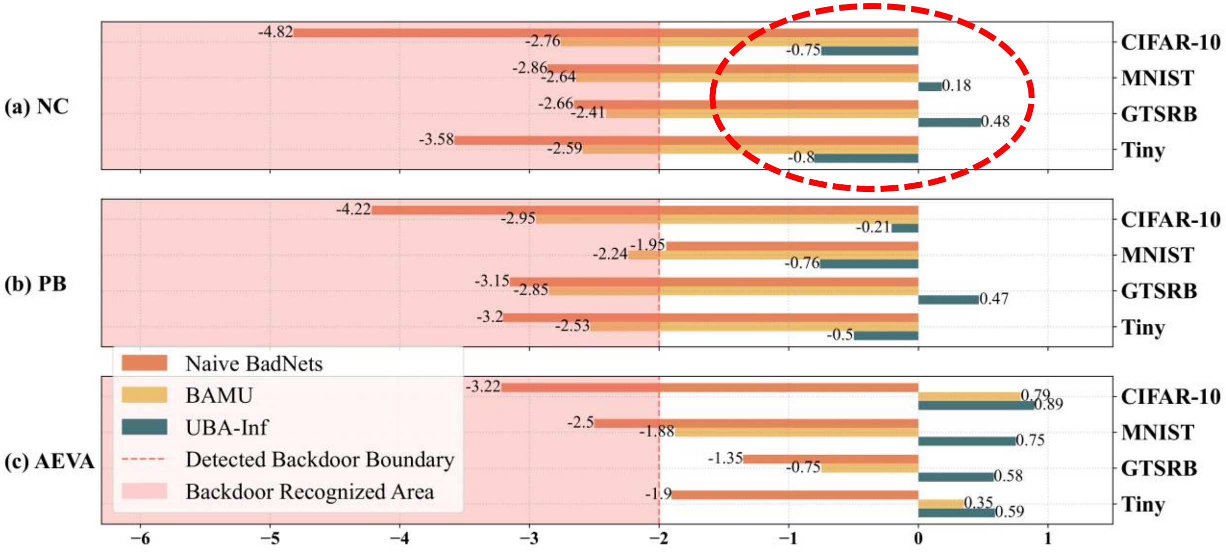
UBA-Inf camouflages the backdoor, and the reversed backdoor is confusing.

- UBA-Inf samples cannot be filtered by popular backdoor sample filters.

- UBA-Inf samples cannot be revealed by model scanners before unlearning with a seemingly normal anomaly score.

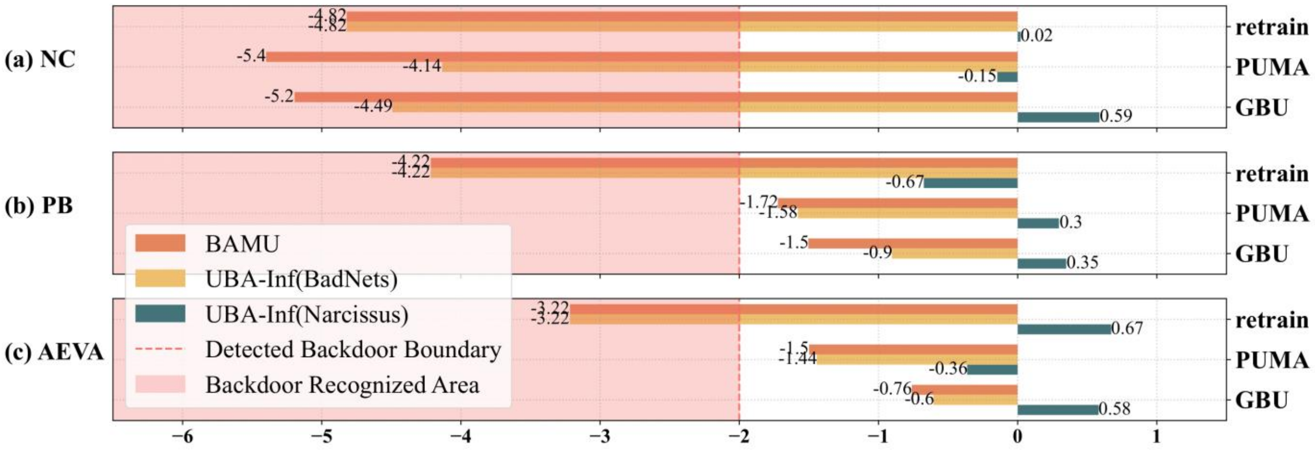
UBA-Inf can confuse different backdoor defenses.



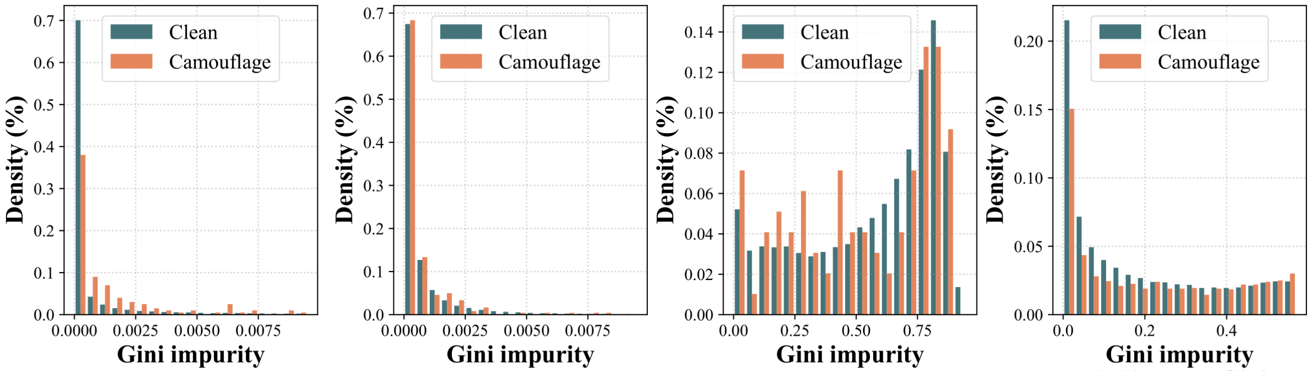


Evaluation: Stealthiness after unlearning & Resistance to reconstruction

- UBA-Inf samples cannot be revealed by model scanners **even after approximate unlearning** with a seemingly normal anomaly score.



- UBA-Inf camouflage samples are confused with normal samples, so unlearning defenses like MU can hardly filter them.



(a) CIFAR-10 MU Gini (b) MNIST MU Gini (c) GTSRB MU Gini (d) Tiny MU Gini

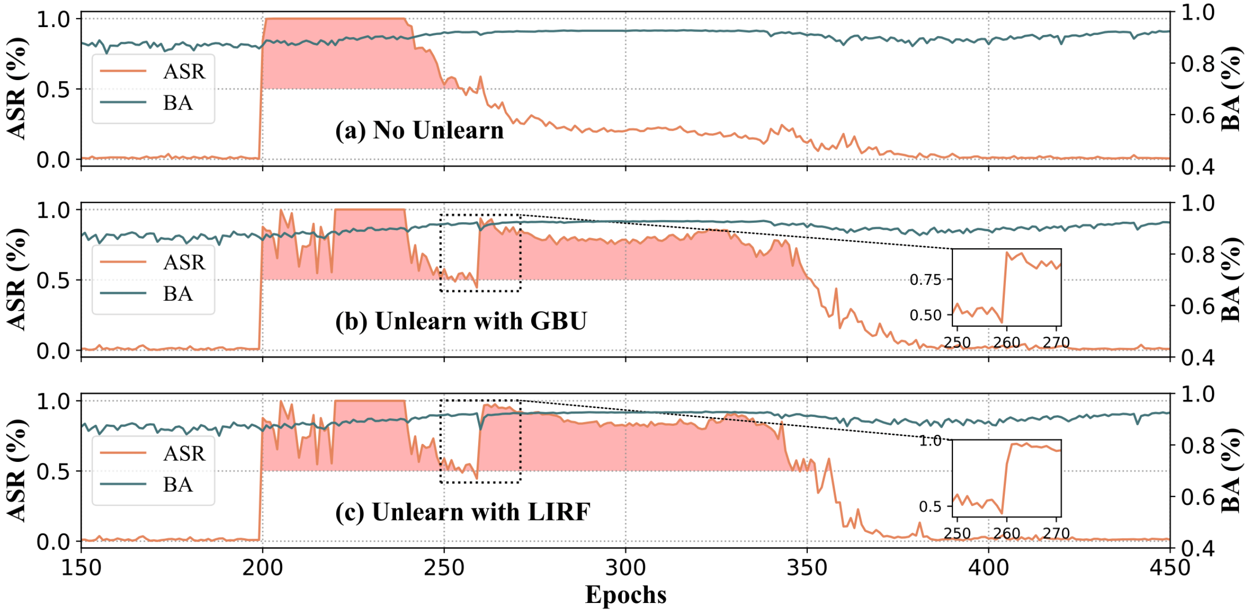
- UBA-Inf can still be activated by unlearning even after model re-construction defenses.

Defenses	before unlearn BA(%)	before unlearn ASR(%)	PUMA unlearn BA(%)	PUMA unlearn ASR(%)	GBU unlearn BA(%)	GBU unlearn ASR(%)
CIFAR-10						
FT	93.28	**8.18**	85.62	***80.44***	85.71	***80.95***
FP	93.18	**5.00**	85.53	***72.68***	86.44	***83.13***
NAD	92.87	**14.87**	86.62	***70.60***	88.06	***87.54***
MNIST						
FT	99.67	**11.05**	99.01	***77.23***	99.09	***89.12***
FP	99.59	**3.49**	98.77	***62.87***	99.00	***99.56***
NAD	99.62	**17.09**	98.59	***79.17***	98.92	***90.46***
GTSRB						
FT	98.20	**11.45**	95.13	***76.93***	95.39	***71.51***
FP	98.31	**9.29**	95.19	***81.57***	95.09	***70.73***
NAD	98.09	**9.80**	95.37	***88.92***	95.38	***65.31***
Tiny						
FT	55.26	**9.12**	50.16	***40.15***	50.01	***43.29***
FP	55.14	**8.54**	50.02	***42.15***	49.95	***45.16***
NAD	55.25	**10.25**	50.11	***44.74***	50.03	***41.63***

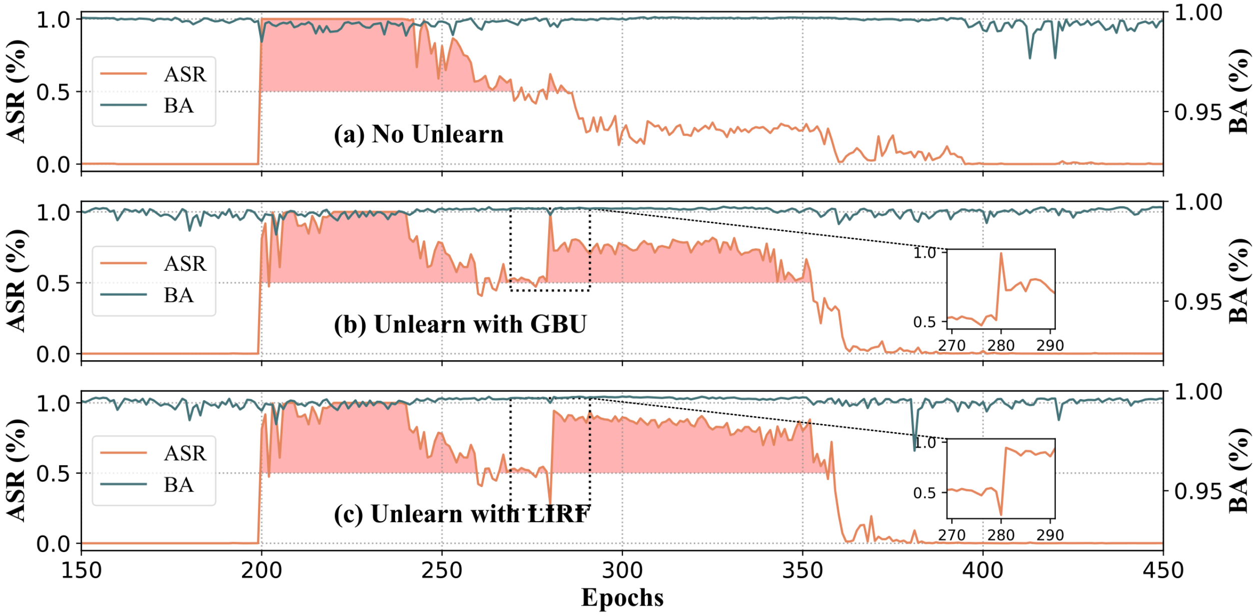
It's disturbing that UBA-Inf can improve backdoor stealthiness and resistance.

Evaluation: Persistence in continuous training

- Assume task datasets in CT-MLaaS are from **either a similar distribution** or different domains in which each task has the same data label space but different feature distributions, a.k.a **Domain-Incremental-Learning**.
- The adversary of UBA-Inf expects the injected backdoor to keep away from backdoor vanishing caused by catastrophic forgetting (**improve backdoor persistence**)



Persistence evaluation on Cifar-10

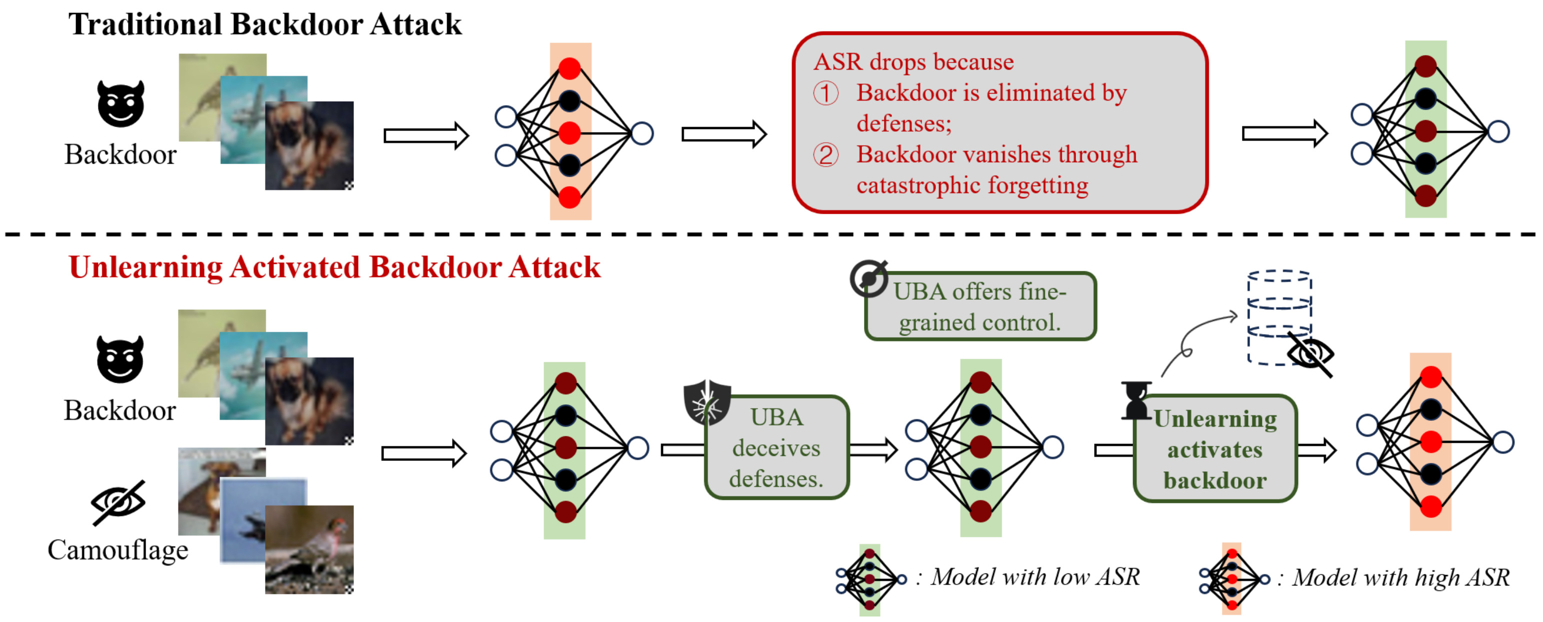


Persistence evaluation on Rotated-MNIST

Conclusion: UBA-Inf achieves 4x persistence improvement with limited poisoning samples (2% of the total training samples).

Conclusion & Take-aways

- *Uncovering vulnerabilities in machine unlearning;*
- *Combining backdoor attacks and unlearning;*
- *Advancing persistent backdoor attacks in continual leaning.*



Thank you!

Q&A

😁 **Contact me:** ***huangzirui@smail.nju.edu.cn***