智能系统安全实践：训练数据隐私&模型窃取

复旦白泽智能

系统软件与安全实验室

大纲

- 模型反演攻击：学习基于数据重建的模型反演攻击
  - 基于输出重建
  - 基于特征重建
  - 基于梯度重建
- 成员推理攻击：了解成员推理攻击相关内容
- 模型窃取攻击：模型蒸馏

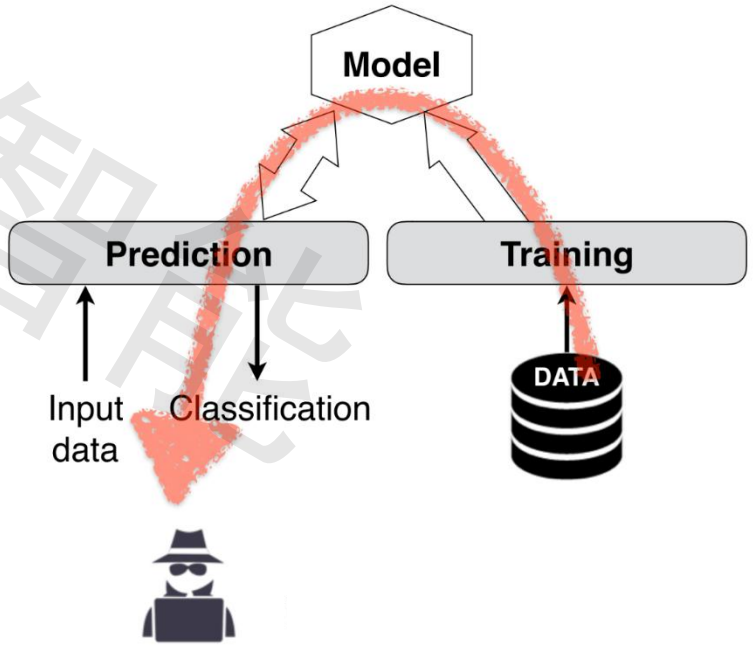
- 实验：
  - 在MNIST上实现针对Lenet5的模型反演攻击
  - 在MNIST上实现针对LeNet5的模型窃取攻击

训练数据隐私

■在服务商发布训练好的模型后，攻击者可以从模型中反推训练数据

■具体方法

- ■模型反演：白盒场景直接逆向数据
- ■成员推理：黑盒场景推理数据从属



黑客拿到了你的数据！

模型反演

■应用场景：

■基于深度学习的人脸识别模型

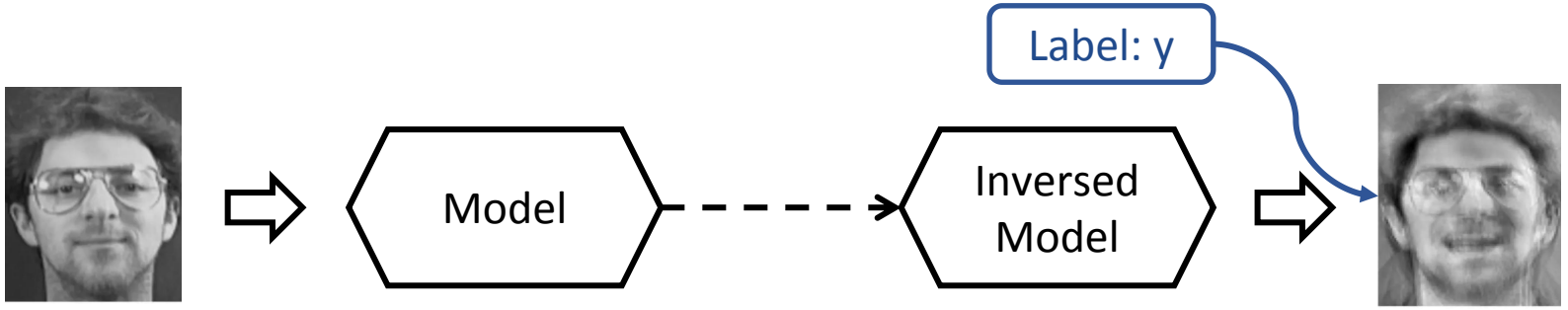
1. 攻击者通过某种方式窃取到模型
2. 根据模型反演得到模型训练集中的图片
3. 即获取到每个人的隐私训练数据（照片）

■语音识别等场景下也存在类似的问题

Training Set　　Recovered Image

■攻击策略：

■找到模型对于某一类别，置信度最高的图片（*“平均脸”恢复*）



模型反演

■攻击目标：

■给定训练好的模型$f_\theta(x)$，利用$f_\theta(x)$反推出可能的训练数据

■优化目标：$\min_x \ell(f_\theta(x), \tilde{y})$ 指定的类别

■存在问题：反推出的图片可能具有很大噪音

■解决方案：引入Total Variation (TV) Loss，使图像更加平滑

■$\ell_{TV}(x) = \frac{1}{M}\sum_{ij}(x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2$ *减小图像中相邻像素之间的差异*

■其中

- i, j表示图像的第i行和第j列
- M为图片的总维度（MNIST->784）

模型反演

■新优化目标：$\min_x \ell(f_\theta(x), \tilde{y}) + \lambda \cdot \ell_{TV}(x)$

■其中 λ 为给定的权重超参数

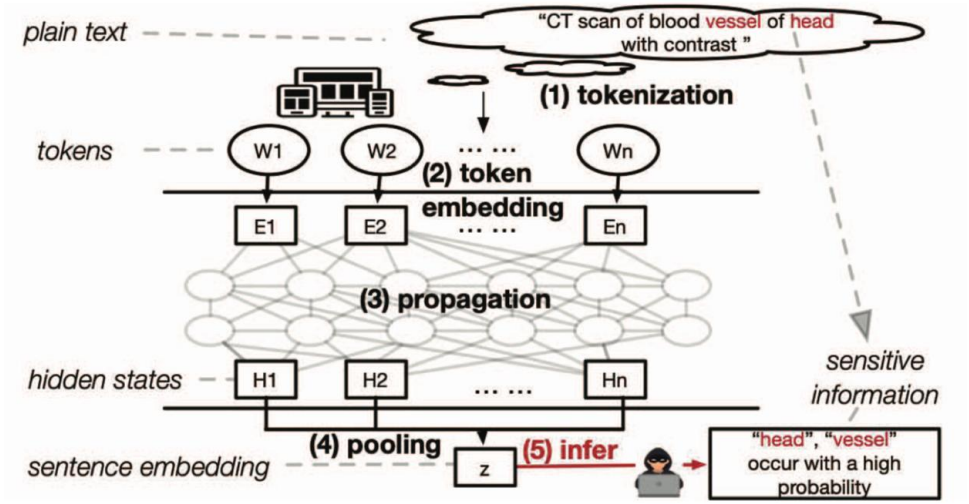
■算法框架：

■给定模型$f_\theta(x)$和目标类别$\tilde{y}$

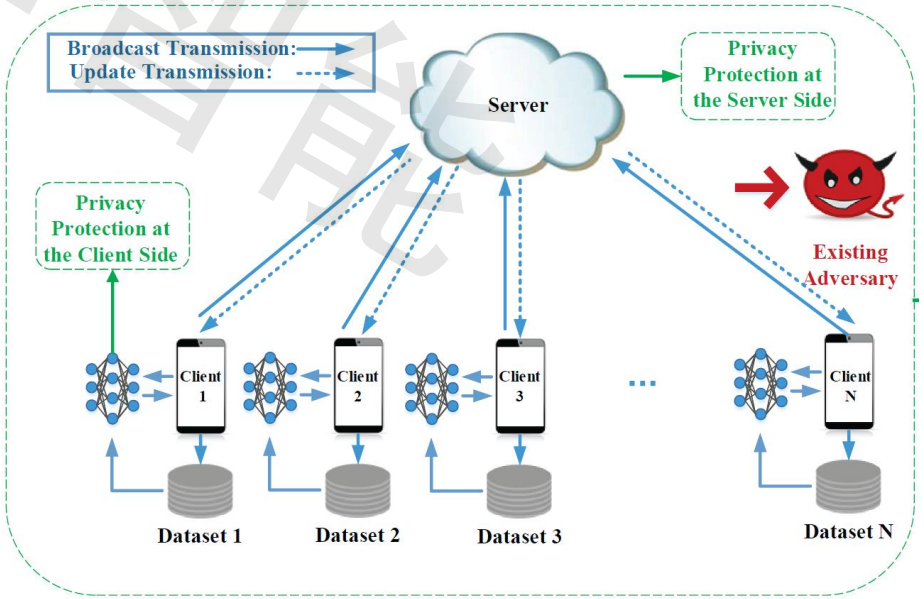
1. 初始化图片x为随机噪声图片
2. 每次迭代中：
   1. 计算损失函数$\ell(f_\theta(x), \tilde{y}) + \lambda \cdot \ell_{TV}(x)$
   2. 固定模型参数θ，利用梯度优化图像x
3. 重复直至模型收敛

其他窃取场景

- 在语言模型的场景中：
  - 攻击者可以劫持模型的特征来反推用户查询的数据
- 在协同训练的场景中：
  - 攻击者（用户之一）可以通过模型返回的梯度信息反推其他用户的数据



语言模型



协同训练

成员推理

■上述窃取场景都基于白盒假设：攻击者可以拿到模型的参数

■成员推理：尝试从部署的服务（API）中窃取训练数据

- ■无需模型参数，仅通过“查询”操作获取信息
- ■应用场景：获取ChatGPT的训练数据信息

■攻击目标：数据是否在训练集中？

Chat-GPT隐私泄漏

成员推理

■攻击思路：

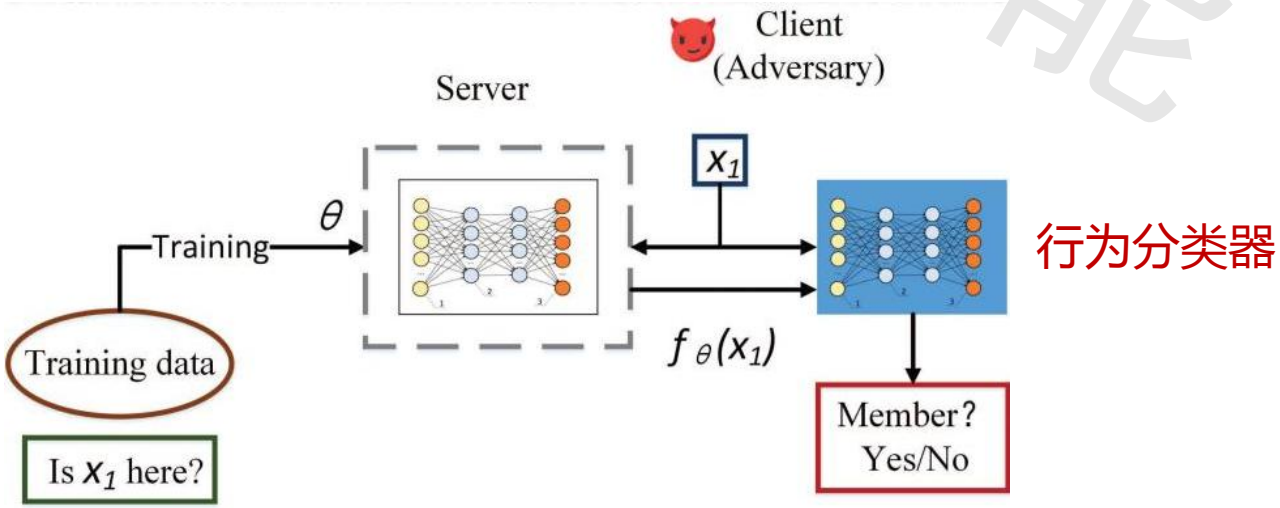
■模型在训练样本与非训练样本上的行为表现不同

■攻击策略：

■通过模型蒸馏，在本地训练一个目标模型的代理模型

■利用代理模型在代理训练集与非训练集上的行为差异训练分类器

■使用训练好的分类器判断目标样本是否存在于目标模型的训练集中



训练数据隐私小结

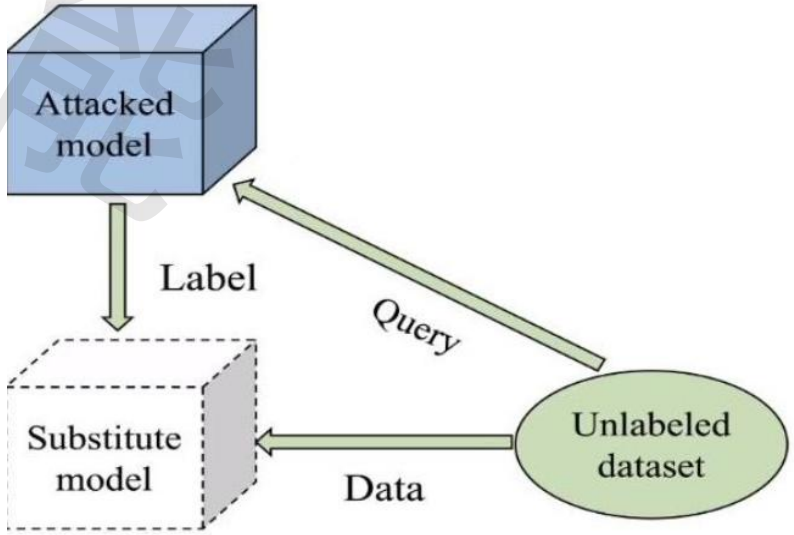
- 模型反演
  - 白盒假设：攻击者可以拿到模型的参数
  - 攻击目标：直接反推出模型训练数据
  - 攻击手段：输出重建、特征重建、梯度重建等

- 成员推理
  - 黑盒假设：攻击者不知道模型的参数
  - 攻击目标：推理出某个数据是否在训练集中
  - 攻击手段：代理模型+行为分类器+预测置信度

模型窃取

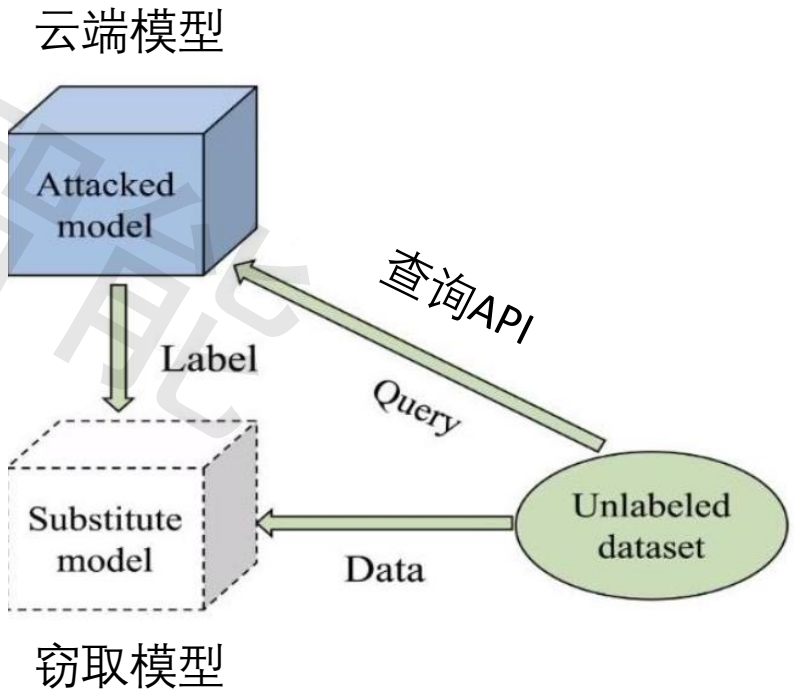
- 模型产权保护
  - 模型API服务的成本高昂
  - GPT-3的训练成本约为140万美元
  - OpenAI每天为运行ChatGPT投入的成本可能高达70万美元

- 模型窃取攻击
  - 攻击者可能窃取训练与部署成本高昂的模型
  - 用于盈利或者其他恶意目的



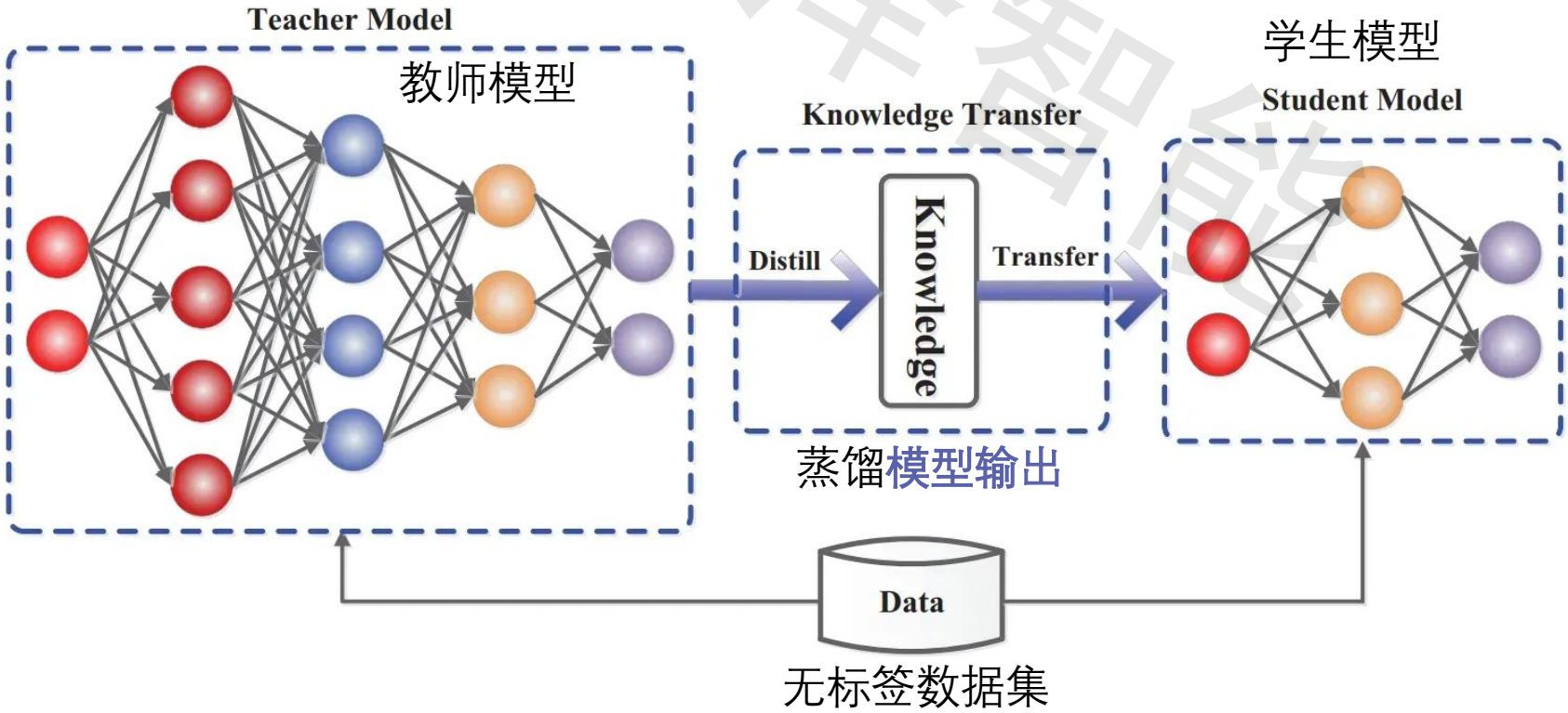
模型窃取

- 攻击限制
  - **黑盒攻击**: 攻击者只能查询模型API
- 攻击目标
  - 向云端模型输入不同的查询样本
  - 训练一个**功能相近**的窃取模型
  - **模型蒸馏**是一种常见的窃取攻击手段



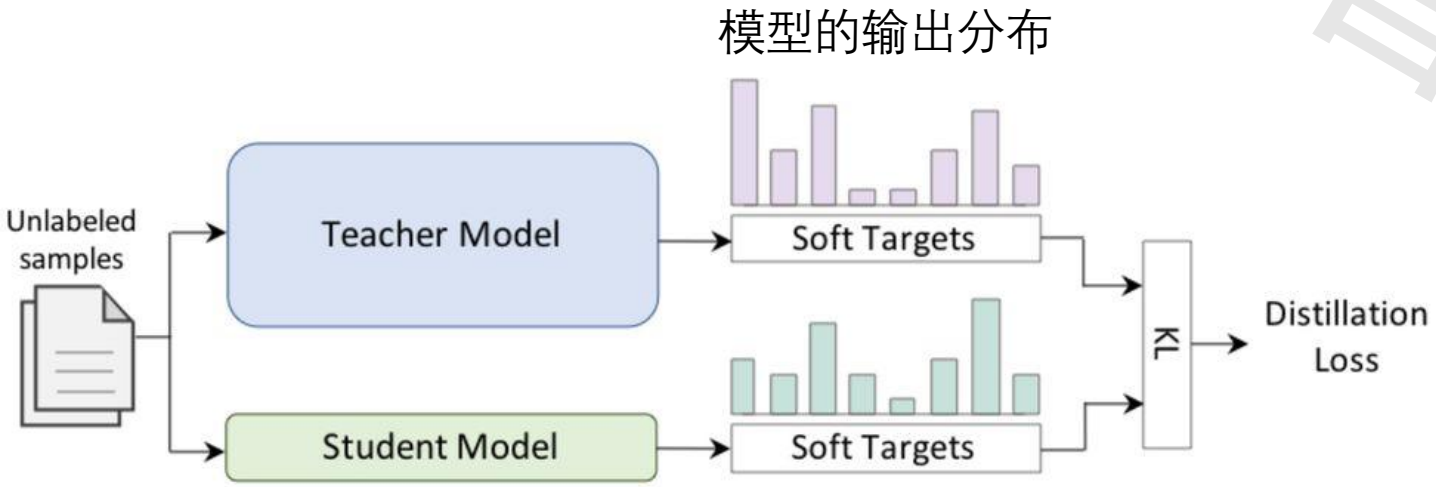
模型蒸馏

- 攻击策略
  - 以云端模型为“**教师模型**”，本地模型为“**学生模型**”
  - 让“学生模型”尽可能**模仿**“教师模型”的行为



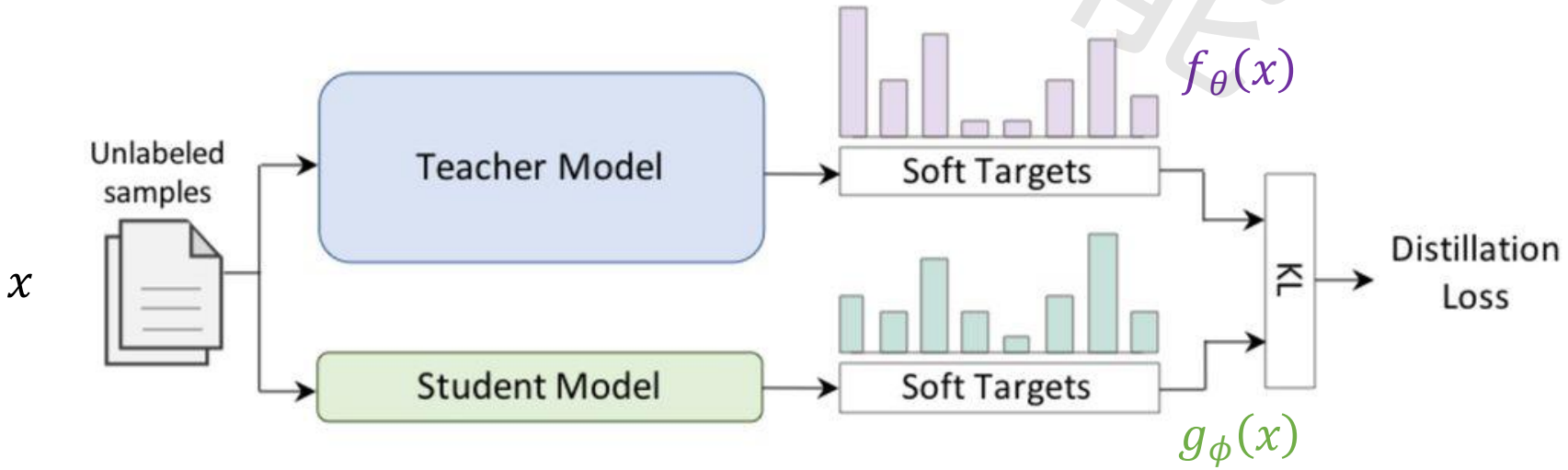
模型蒸馏

- 为什么模仿 == 更小的训练代价？
  - **模型输出**蕴含着比标签更丰富的信息
  - 同时不需要人工对图片做**标注**



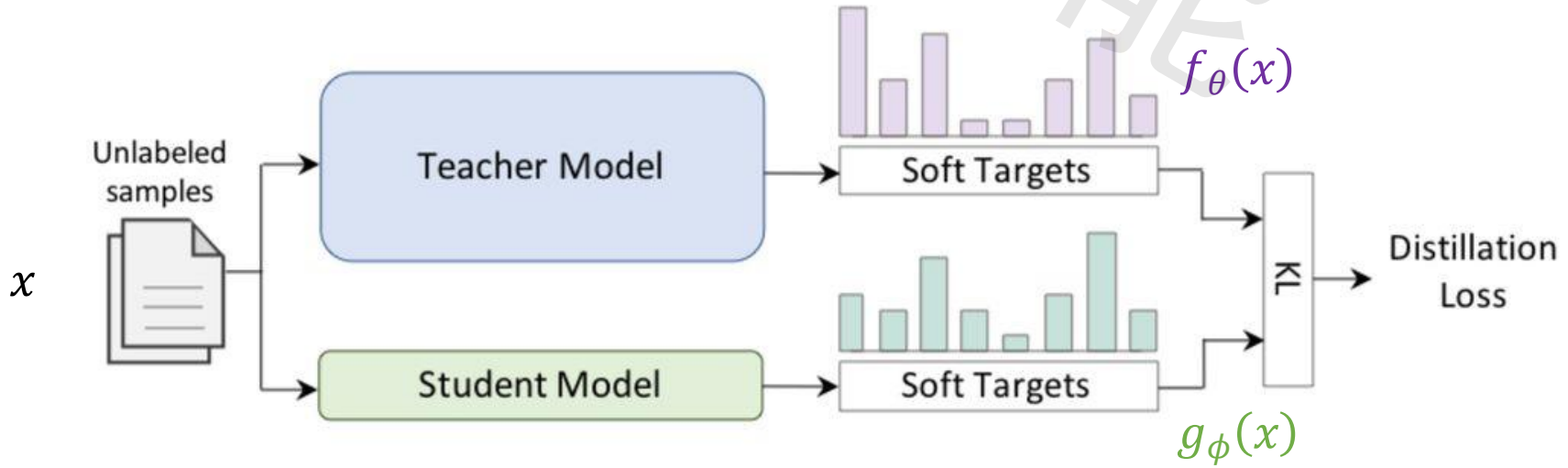
模型蒸馏

- 攻击策略
  - 对于一个**无标签**样本x，
  - 输入到云端模型，得到输出$f_\theta(x)$
  - 输入到本地模型，得到输出$g_\phi(x)$
  - 训练$g_\phi(x)$，让$g_\phi(x)$与$f_\theta(x)$尽可能接近



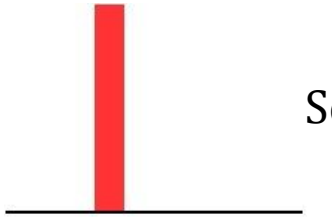
模型蒸馏

- 第一种实现
  - 利用**平方差损失函数**实现窃取目标
  - $\ell(\phi) = \|f_\theta(x) - g_\phi(x)\|_2$
  - 梯度下降更新本地模型的参数ϕ
  - 不需要数据的真实标签即可完成模型训练

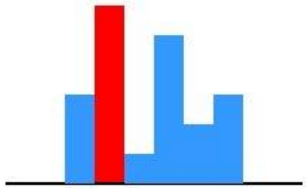


模型蒸馏

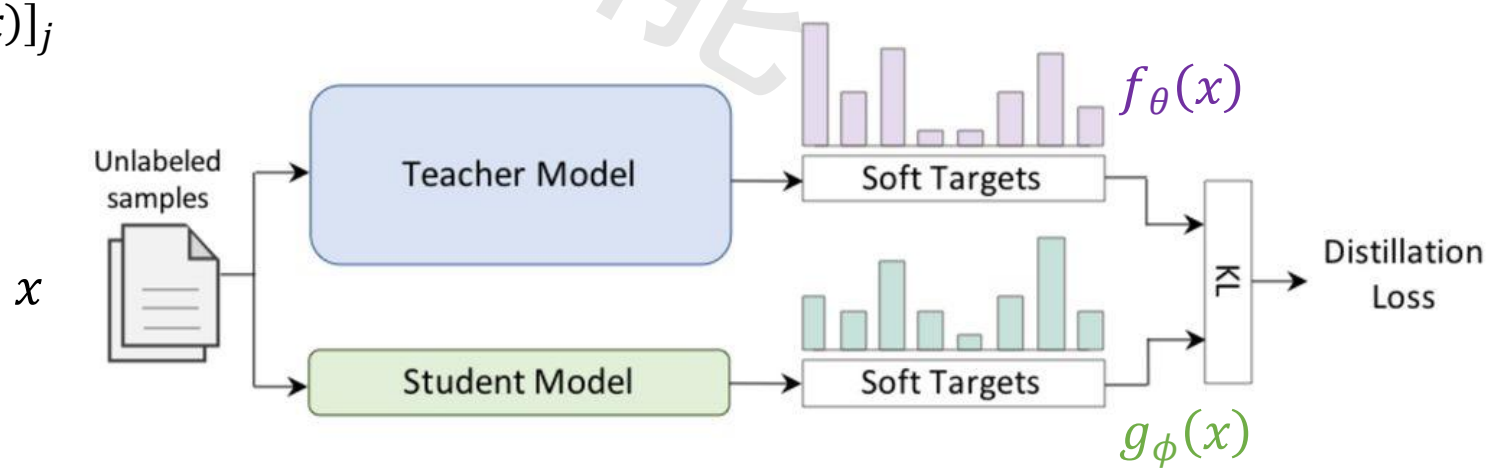
- 第二种实现
  - 令$\tilde{f}_\theta(x) = \text{Softmax}\left(\frac{f_\theta(x)}{\tau}\right)$，$\tilde{g}_\phi(x) = \text{Softmax}(g_\phi(x))$
  - 此处τ将其他类的置信度分数放大
  - 然后利用**KL散度**作为损失函数
  - $\ell(\phi) = \sum_j [\tilde{f}_\theta(x)]_j \log \frac{[\tilde{f}_\theta(x)]_j}{[g_\phi(x)]_j}$



$\text{Softmax}(f_\theta(x))$



$\text{Softmax}\left(\frac{f_\theta(x)}{\tau}\right)$



Q&A

实验内容：基于输出重建的模型反演

- 在MNIST上实现针对Lenet5的模型反演攻击
  - 给定的已训练好的Lenet5模型；
  - 对于该模型的每个类别，实现基于输出重建的模型反演；
  - 结果验证：
    - 调整TV Loss的超参数λ、相似性损失、初始化方式等细节，选择最佳重建效果
    - 重建出的图像可被模型正确分类；
    - 可视化重建的结果，与原始图像相似；

实验内容：模型蒸馏

- 模型蒸馏
  - 在MNIST上训练一个**目标**LeNet5模型 (云端模型)
  - 另外初始化一个**本地**LeNet5模型蒸馏 (本地模型)
  - 利用少量数据对目标模型查询输出，然后训练本地模型
  - 尝试模型蒸馏的**两种实现方式**
  - 观察**不同查询数据量**下本地模型的预测准确度

Q&A