Data-Free Knowledge Extraction from Deep Neural Networks

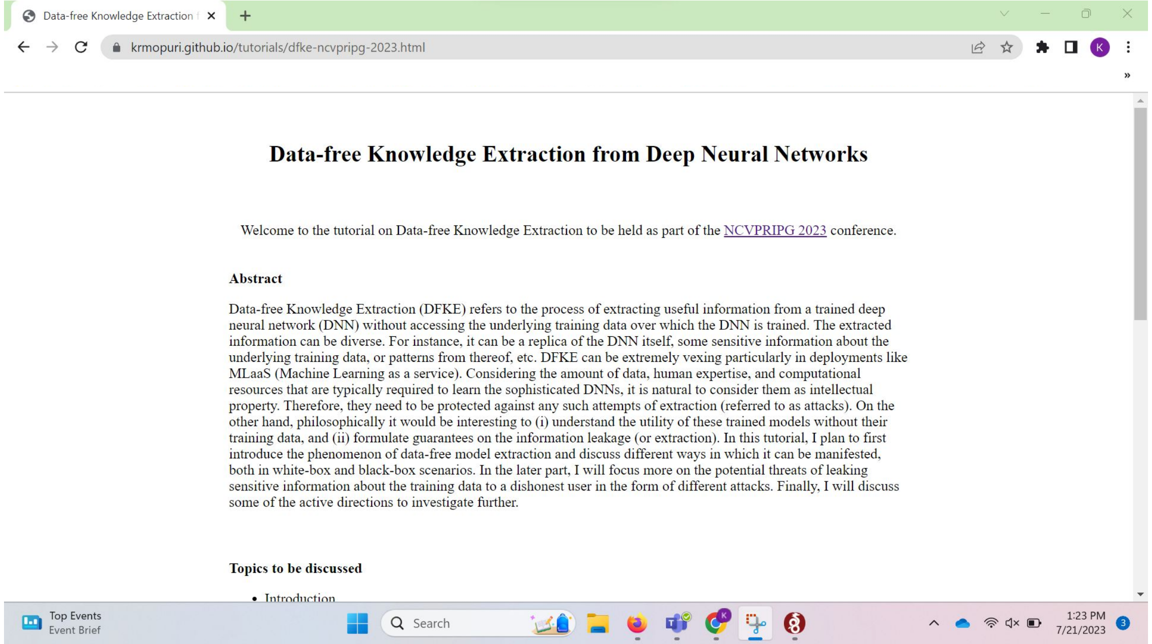
Konda Reddy Mopuri
Dept. of AI, IIT Hyderabad

NCVPRIPG-2023
21-23 July, IIT Jodhpur

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

DiL
Data-driven Intelligence & Learning Lab

Tutorial website



Data-free Knowledge Extraction from Deep Neural Networks

Welcome to the tutorial on Data-free Knowledge Extraction to be held as part of the NCVPRIPG 2023 conference.

Abstract

Data-free Knowledge Extraction (DFKE) refers to the process of extracting useful information from a trained deep neural network (DNN) without accessing the underlying training data over which the DNN is trained. The extracted information can be diverse. For instance, it can be a replica of the DNN itself, some sensitive information about the underlying training data, or patterns from thereof, etc. DFKE can be extremely vexing particularly in deployments like MLaaS (Machine Learning as a service). Considering the amount of data, human expertise, and computational resources that are typically required to learn the sophisticated DNNs, it is natural to consider them as intellectual property. Therefore, they need to be protected against any such attempts of extraction (referred to as attacks). On the other hand, philosophically it would be interesting to (i) understand the utility of these trained models without their training data, and (ii) formulate guarantees on the information leakage (or extraction). In this tutorial, I plan to first introduce the phenomenon of data-free model extraction and discuss different ways in which it can be manifested, both in white-box and black-box scenarios. In the later part, I will focus more on the potential threats of leaking sensitive information about the training data to a dishonest user in the form of different attacks. Finally, I will discuss some of the active directions to investigate further.

Topics to be discussed

- Introduction

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

DiL
Data-driven Intelligence
& Learning Lab

1

- Introduction
  - DFKE
  - How it happened to me
  - Noise optimization for DFKE)
    1. Preliminaries (KD, etc.)

2

- Generative Reconstruction for DFKE
  - Preliminaries (GANs, etc.)
- Adversarial Exploration for DFKE

3

- Data-free Attacks
- Conclusion

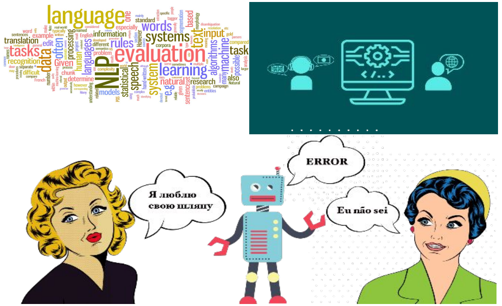
1

Introduction

How it all started!

Success of Deep Learning

- Numerous applications
- Impressive performances



Apologies to the sources of the pictures, lost them in copying across my slides

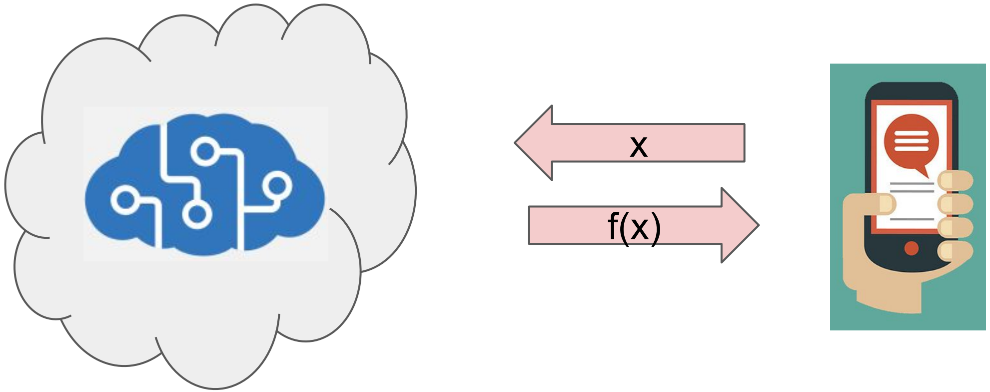
Deployment

Deployment

1. Handing Over the model physically

Deployment

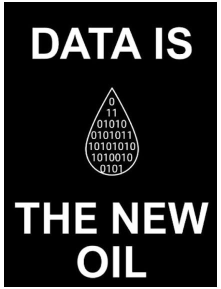
2. Allowing (pay-per-query) access over the cloud (MLaaS)

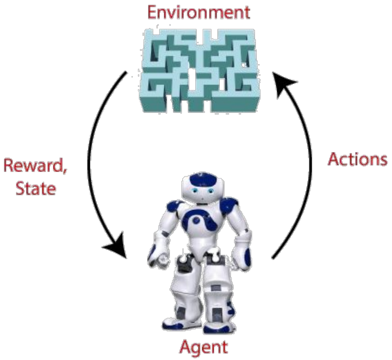


Handing over the model physically

Absence of training data (?!) 

- We may have the trained models but not the training data





 భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Apologies to the sources of the pictures, lost them in copying across my slides

DiL
Data-driven Intelligence & Learning Lab

Models in the absence of training data

- Can
  - Inference (deploying)
  - Pre-training and Transfer Learning
- Can't (?)
  - Compression & Distillation
  - Fine-tuning & Continual learning
  - Adapting, etc.

Knowledge Distillation

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Data-driven Intelligence & Learning Lab

Knowledge Distillation (KD)

- High-capacity **Teacher** model → a smaller **Student** model

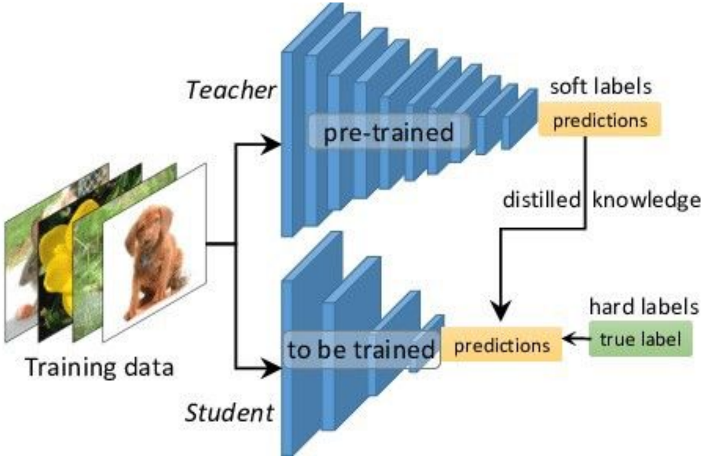


Figure from https://towardsdatascience.com

Knowledge Distillation (KD)

Useful for Model Compression

- High-capacity **Teacher** model → a smaller **Student** model

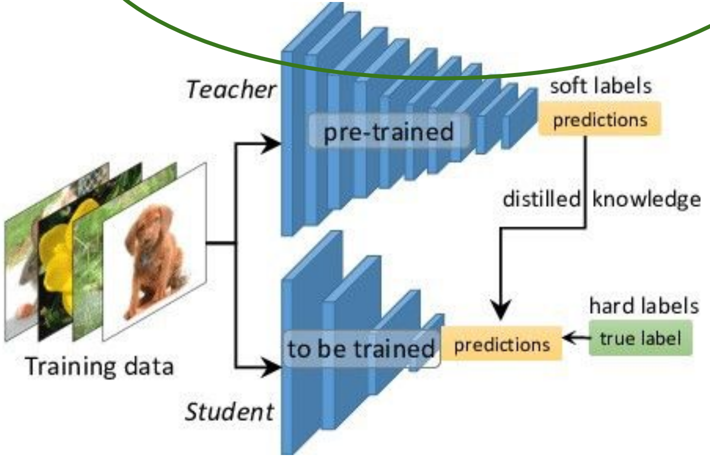
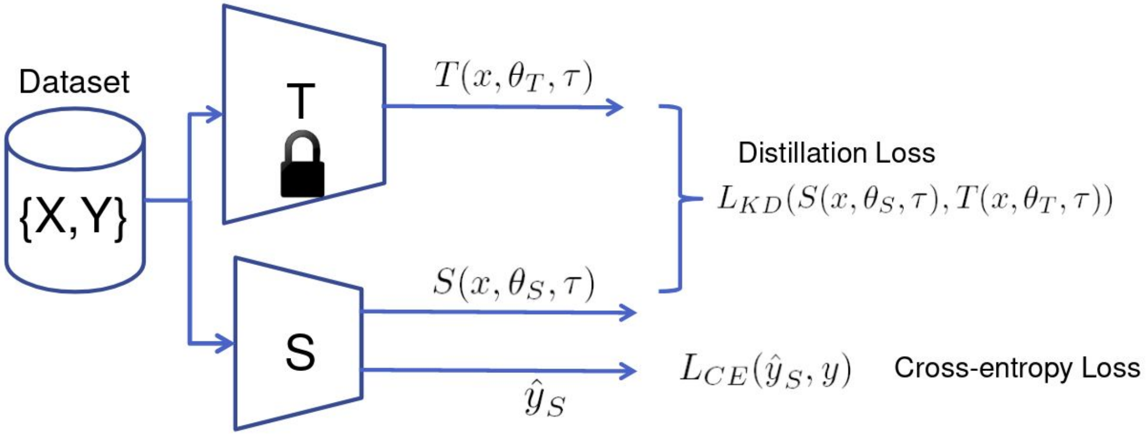


Figure from https://towardsdatascience.com

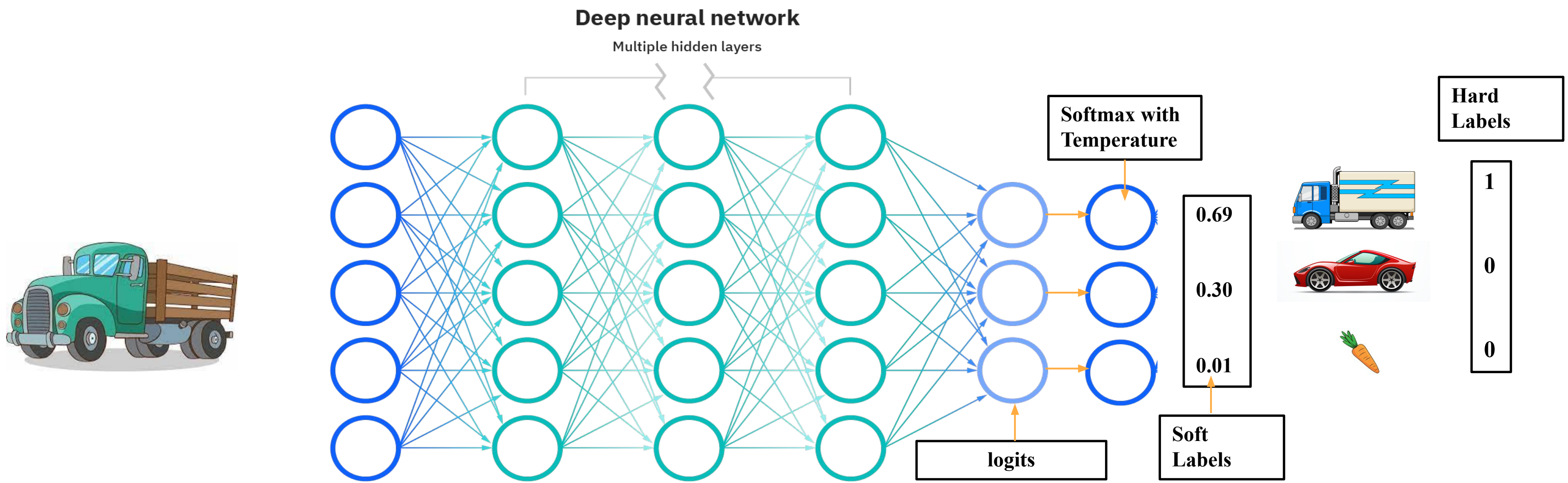
Knowledge Distillation (KD)



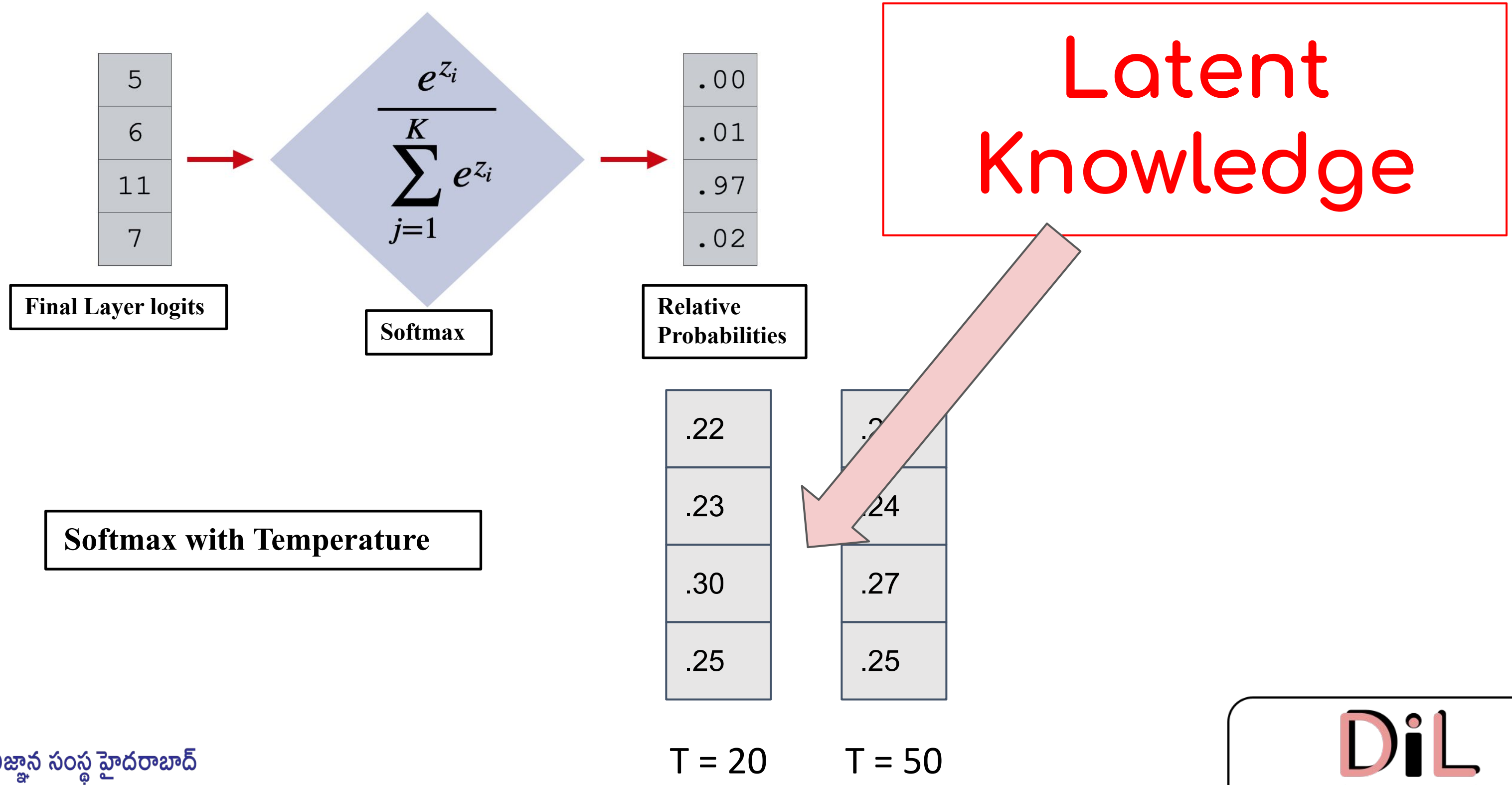
$$L = \sum_{(x,y)\in\mathbb{D}} L_{KD}(S(x, \theta_S, \tau), T(x, \theta_T, \tau)) + \lambda L_{CE}(\hat{y}_S, y)$$

Hinton et al. Distilling the Knowledge in a Neural Network, 2015

Knowledge Distillation (KD)

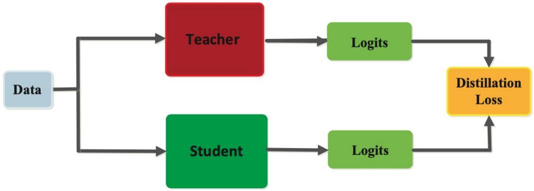


Knowledge Distillation (KD)

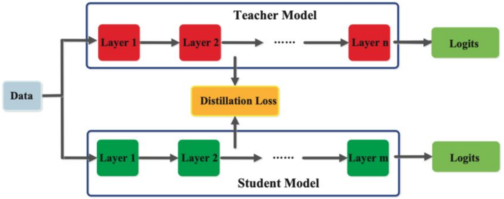


Knowledge Distillation (KD) - types

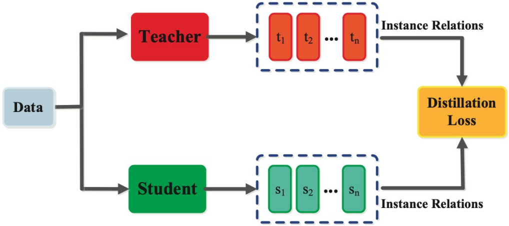
- Prediction/Response Distillation



- Feature Distillation



- Relation Distillation



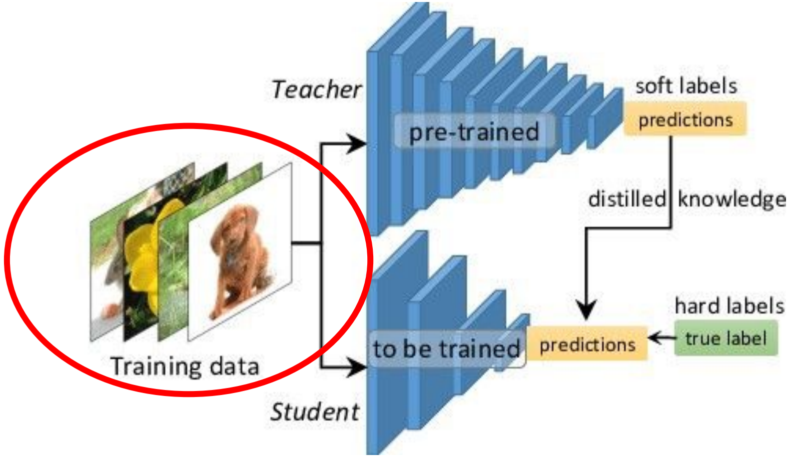
భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Survey on KD
IJCV 2021

Data-driven Intelligence & Learning Lab

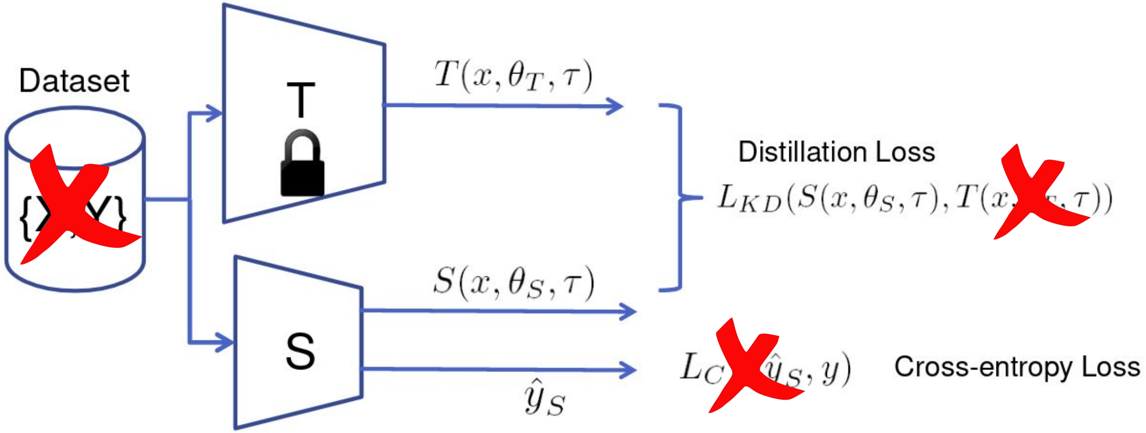
Requirement

Transfer set

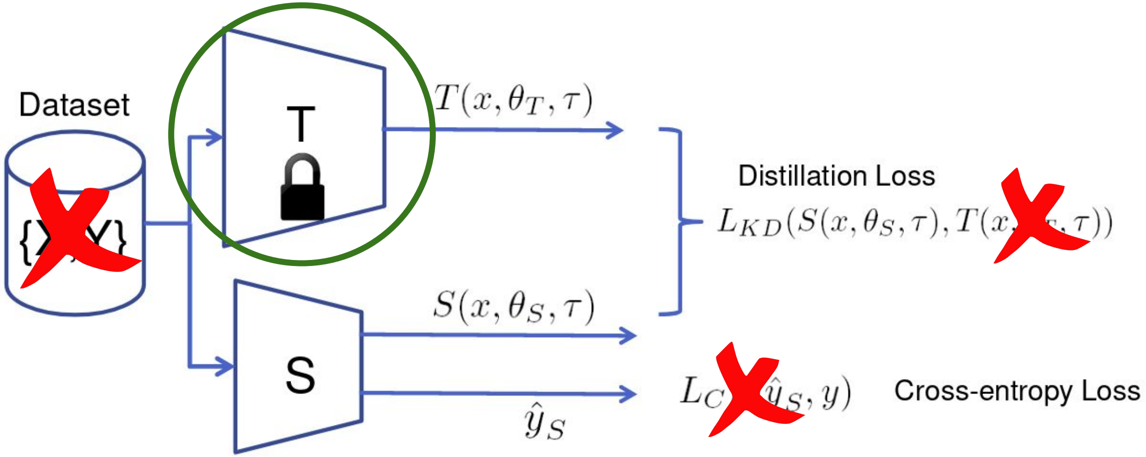


Requires Training Data on which T is trained

KD in the absence of training data



KD in the absence of training data



Can the trained Teacher model help with transfer set?

Mining Data-Impressions from Deep Models as Substitute for Unavailable Training Data

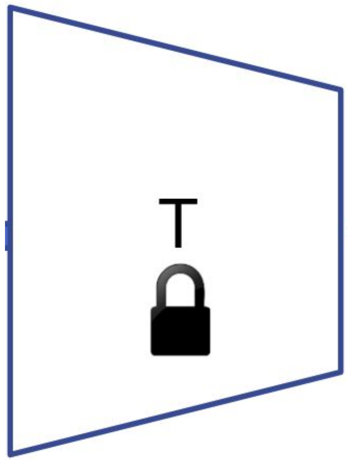
Konda Reddy Mopuri et al.

ICML 2019 & Trans. on PAMI 2021

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

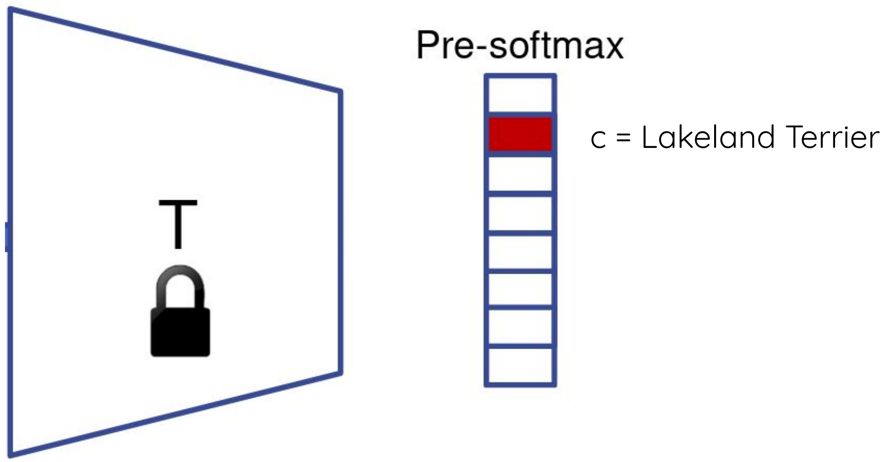
DiL
Data-driven Intelligence & Learning Lab

Class Impressions: Parameters → patterns



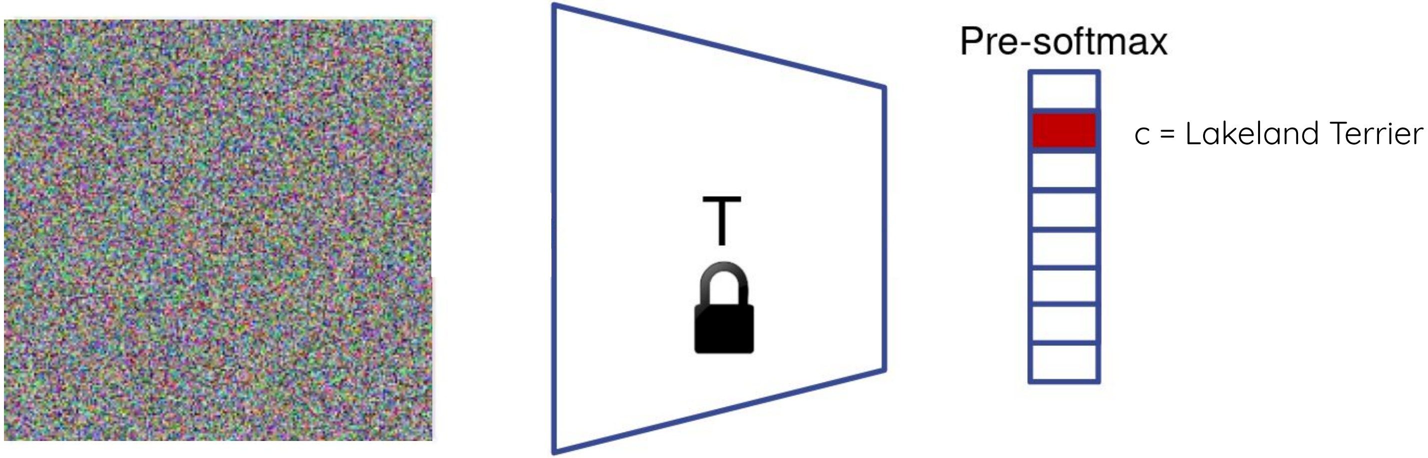
Konda Reddy Mopuri et al., Ask, Acquire and Attack: Data-free UAP generation using Class impressions, ECCV 2018

Class Impressions: Parameters → patterns



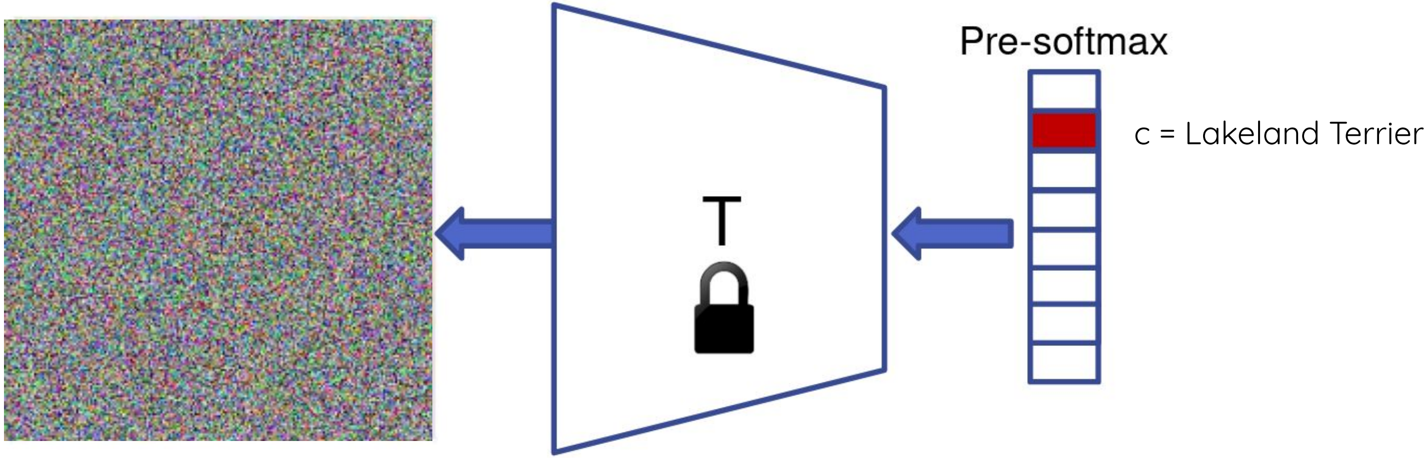
Konda Reddy Mopuri et al., Ask, Acquire and Attack: Data-free UAP generation using Class impressions, ECCV 2018

Class Impressions: Parameters → patterns



Konda Reddy Mopuri et al., Ask, Acquire and Attack: Data-free UAP generation using Class impressions, ECCV 2018

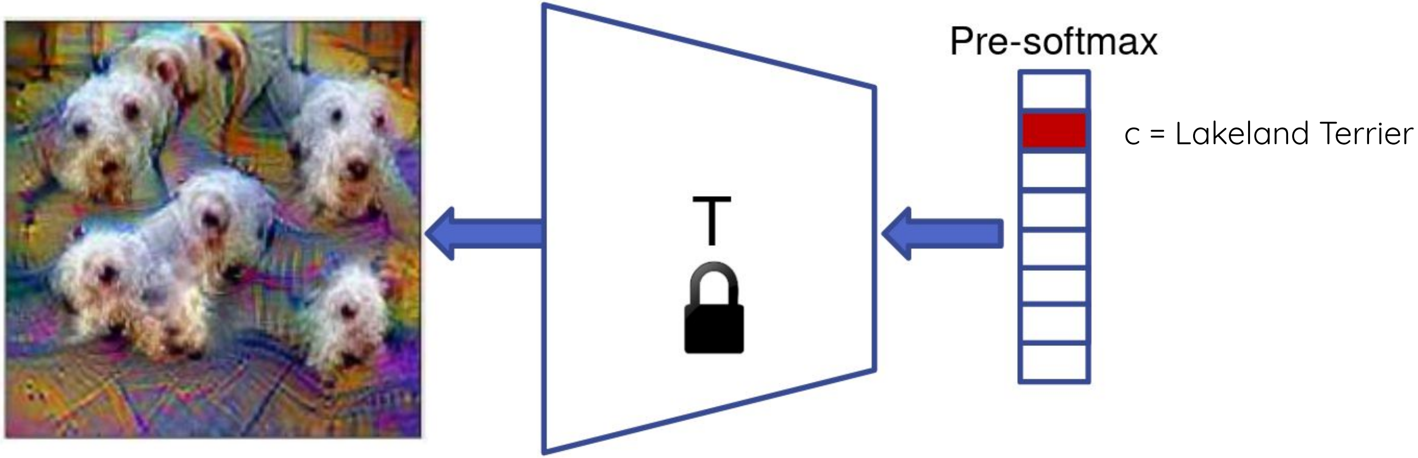
Class Impressions: Parameters → patterns



$$CI_c = argmax_x \ \ T_c(x)$$

Konda Reddy Mopuri et al., Ask, Acquire and Attack: Data-free UAP generation using Class impressions, ECCV 2018

Class Impressions: Parameters → patterns



$$CI_c = argmax_x \; T_c(x)$$

Konda Reddy Mopuri et al., Ask, Acquire and Attack: Data-free UAP generation using Class impressions, ECCV 2018

Class Impressions: Parameters → patterns

Goldfish Cock Wolf spider Lakeland terrier Monarch

Konda Reddy Mopuri et al., Ask, Acquire and Attack: Data-free UAP generation using Class impressions, ECCV 2018

Training on CIs: Limitations

- Generated samples are less faithful and diverse
- One-hot vector labels are reconstructed
  - → minimal latent/dark knowledge → not so close to the natural data
- Student suffers poor generalization

Need an Improved modelling of the output space

Dirichlet modelling of output space

- Softmax space of each class ‘k’ $y^k \sim Dir(K, \alpha^k)$
- Support is the probabilities of a **K**-way classification
- Concentration param (α) → spread of the distribution

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

DiL
Data-driven Intelligence & Learning Lab

Dirichlet modelling of output space

$$y^k \sim Dir(K, \alpha^k)$$

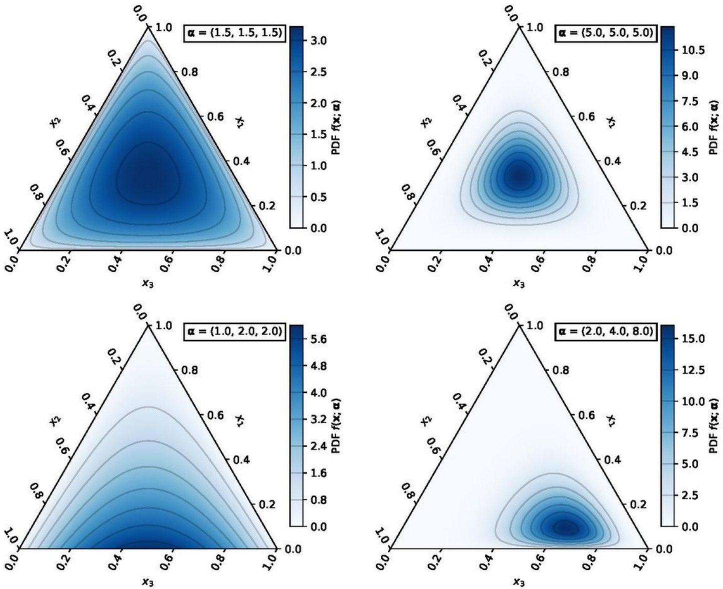


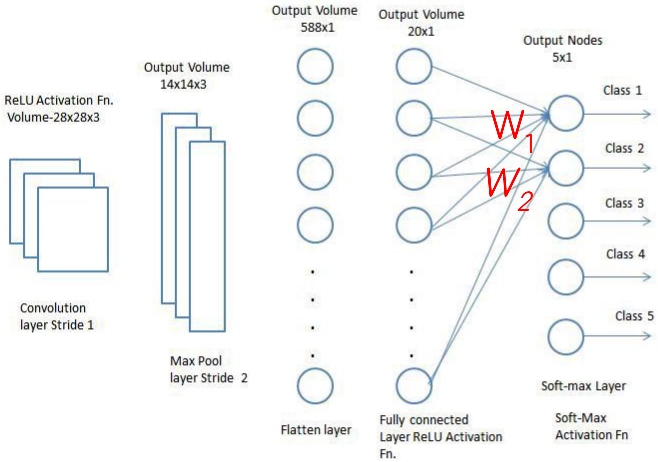
Figure credits: Wikipedia

Dirichlet modelling of output space

- Concentration param (α)
  - Encodes the preferences over the regions of the support
- Samples should reflect the desired inter-class similarities (latent knowledge)

Dirichlet modelling of output space

- Concentration param (α) → inter-class similarities

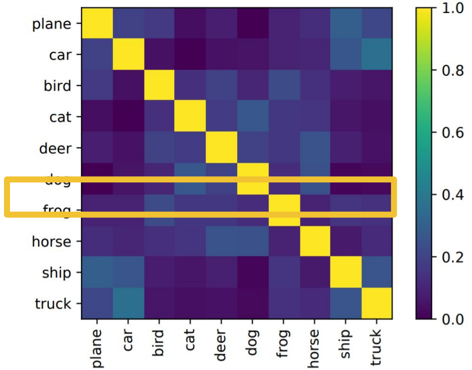


Dirichlet modelling of output space

- Concentration param (α) → inter-class similarities

$$C(i,j) = \frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_j\|}$$

W_k - weights learned by the Teacher's softmax classifier for class 'k'

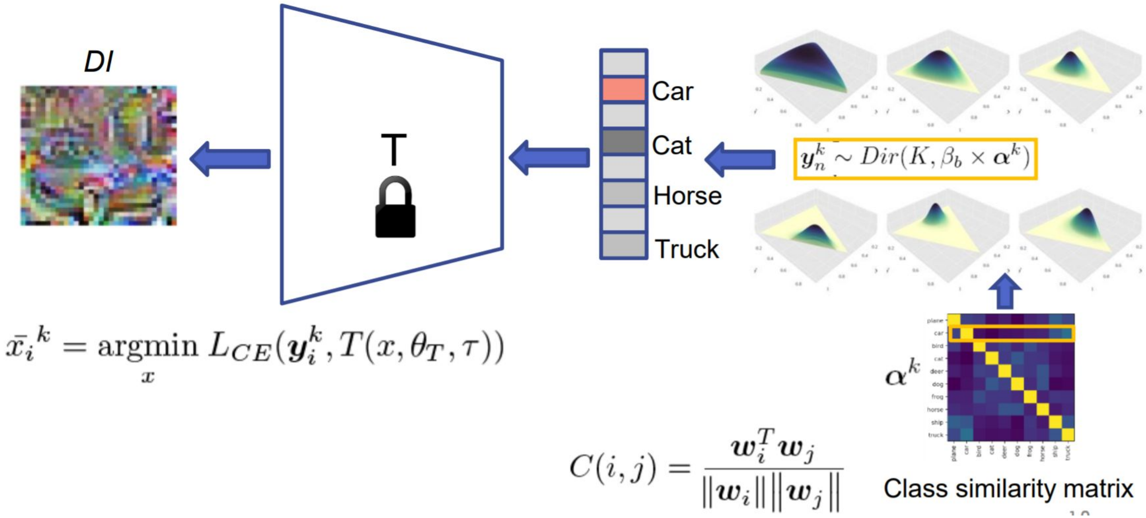


Class similarity matrix

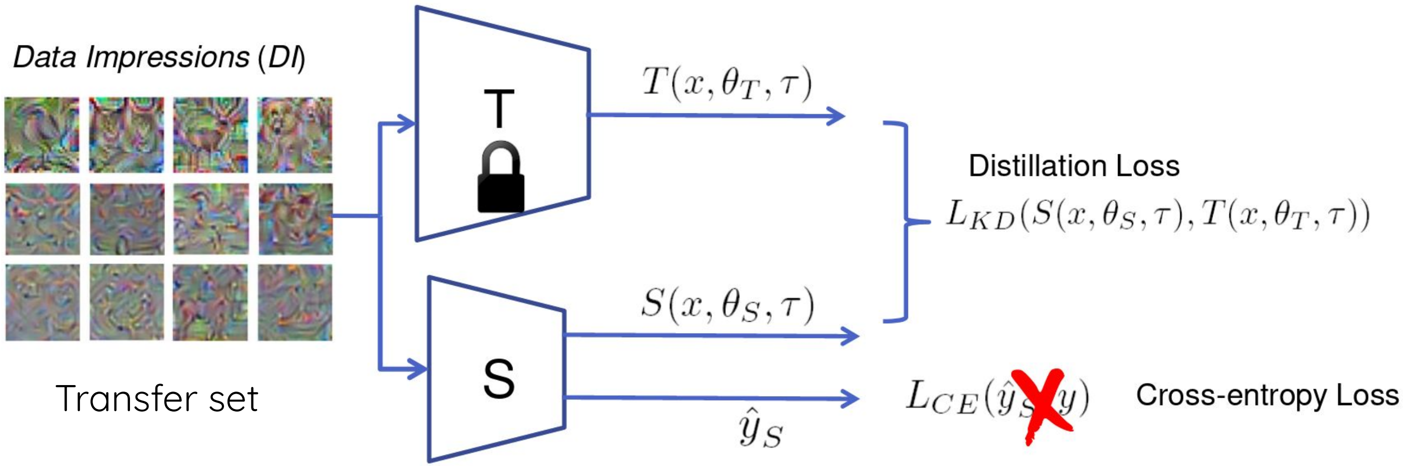
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Data-driven Intelligence & Learning Lab

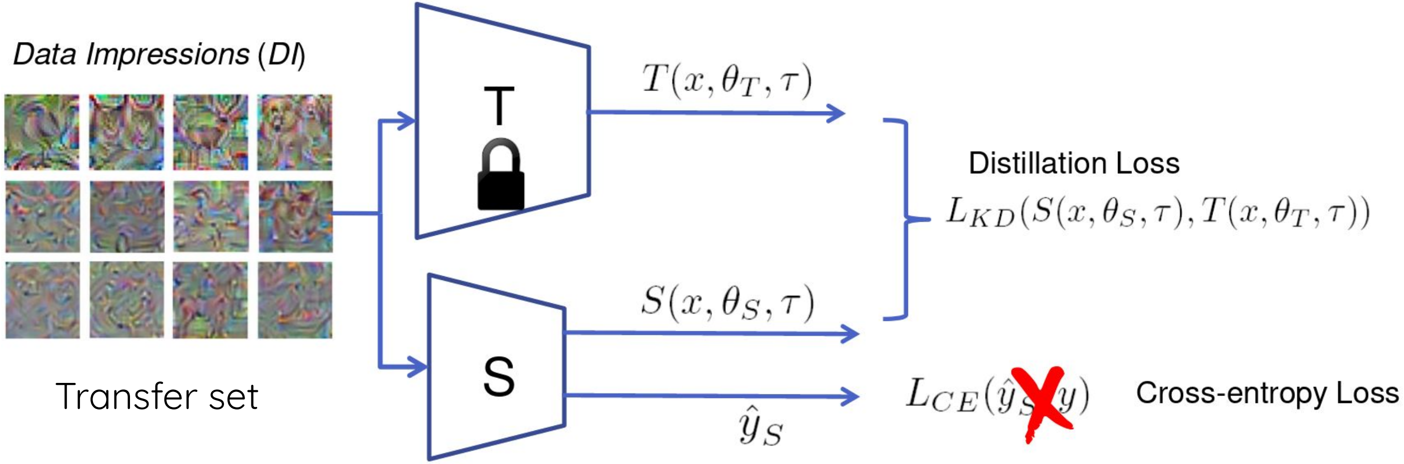
Data Impressions



Distillation with DIs

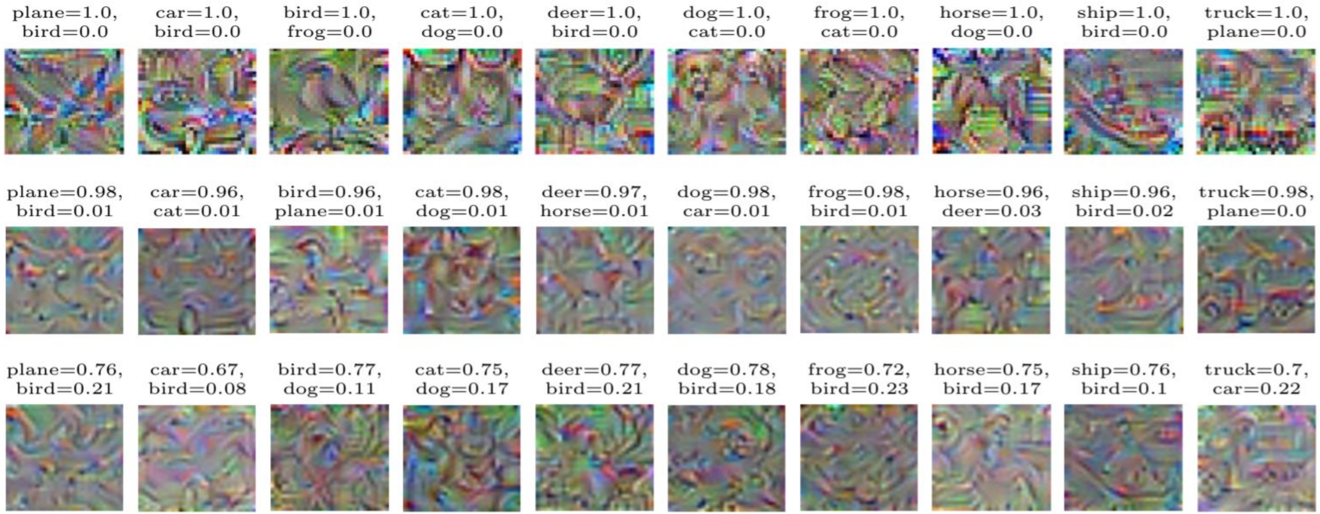


Distillation with DIs



$$\theta_S = \underset{\theta_S}{\operatorname{argmin}} \sum_{\bar{x} \in \bar{X}} L_{KD}(T(\bar{x}, \theta_T, \tau), S(\bar{x}, \theta_S, \tau))$$

Generated Samples



Performance

Model	Performance
Teacher – CE	99.34
Student – CE	98.92
Student–KD (Hinton et al., 2015) 60K original data	99.25
(Kimura et al., 2018) 200 original data	86.70
(Lopes et al., 2017) (uses meta data)	92.47
ZSKD (Ours) (24000 *DI*s, and no original data)	98.77

MNIST

Model	Performance
Teacher – CE	83.03
Student – CE	80.04
Student – KD (Hinton et al., 2015) 50K original data	80.08
ZSKD (Ours) (40000 *DI*s, and no original data)	69.56

CIFAR-10

Performance

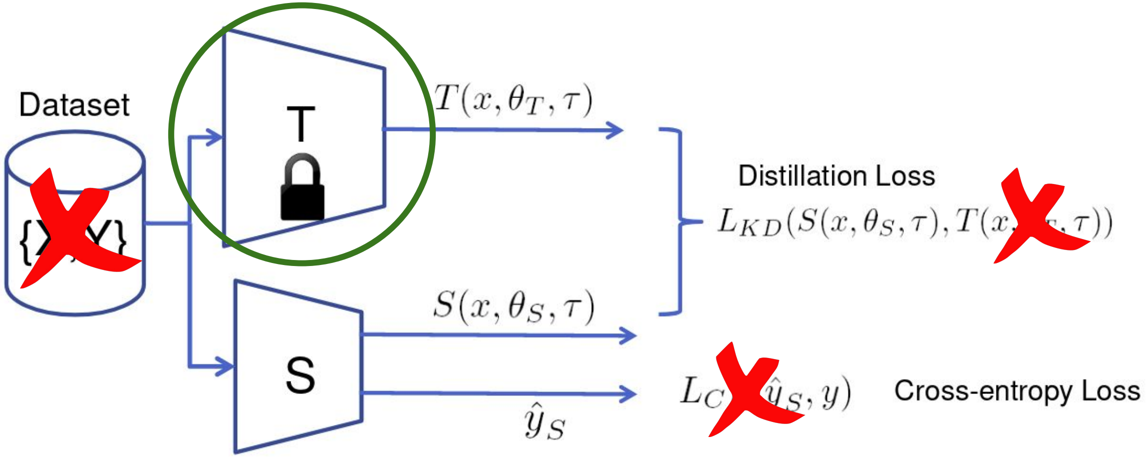
Model	Data-free	Performance (%)
VGG-19 (T)	✗	87.99
VGG-11 (S)- CE	✗	84.19
VGG-11 (S)- KD [9]	✗	84.93
VGG-11 (S)- KD (**Ours**)	✓	74.10
Resnet-18 (S) -CE	✗	84.45
Resnet-18 (S) -KD [9]	✗	86.58
Resnet-18 (S) -KD (**Ours**)	✓	74.76

Model	Data-free	Performance (%)
Resnet-18 (T)	✗	86.54
Resnet-18-half (S)- CE	✗	85.51
Resnet-18-half (S)- KD [9]	✗	86.31
Resnet-18-half (S)- KD (Ours)	✓	81.10

CIFAR-10

Noise Optimization

KD in the absence of training data



Can the trained Teacher model help with transfer set?

Noise Optimization

1. Sample noise (e.g. from a Gaussian distribution)
2. Iterative Gradient Ascent/Descent → Alternate transfer set
3. Perform KD

Noise Optimization

1

$$\tilde{x}^* = \arg\min_{\tilde{x}} \quad \mathcal{R}(\tilde{x}, T)$$

R is the regularization that constraints (prior)

2

$$\tilde{x}^* = \arg\min_{\tilde{x}} \quad \mathcal{R}(\tilde{x}, T) + \mathcal{L}_{CE}(T(\tilde{x}), \tilde{y})$$

Class-conditional transfer sample
Cross-entropy loss

Noise Optimization

$$\arg\min_{S} \sum_{(\tilde{x},\tilde{y})}^{(\tilde{X},\tilde{Y})} \mathcal{L}_{CE}(S(\tilde{x}),\tilde{y}) + \mathcal{L}_{KD}(S(\tilde{x}),T(\tilde{x}))$$

3

Perform the distillation

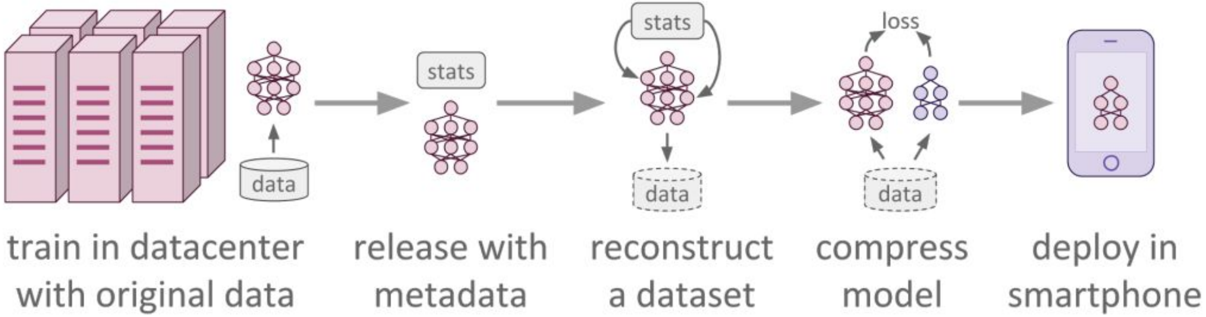
Noise Optimization

$$\tilde{x}^* = \arg\min_{\tilde{x}} \ \mathcal{R}(\tilde{x}, T) + \mathcal{L}_{CE}(T(\tilde{x}), \tilde{y})$$

Suitable regularization for distilling the knowledge from Teacher

Noise Optimization - Regularizing activation

- Lopes et al. 2018 save the activation summary of the Teacher's layers
    - Mean and Variance



భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Data-driven Intelligence & Learning Lab

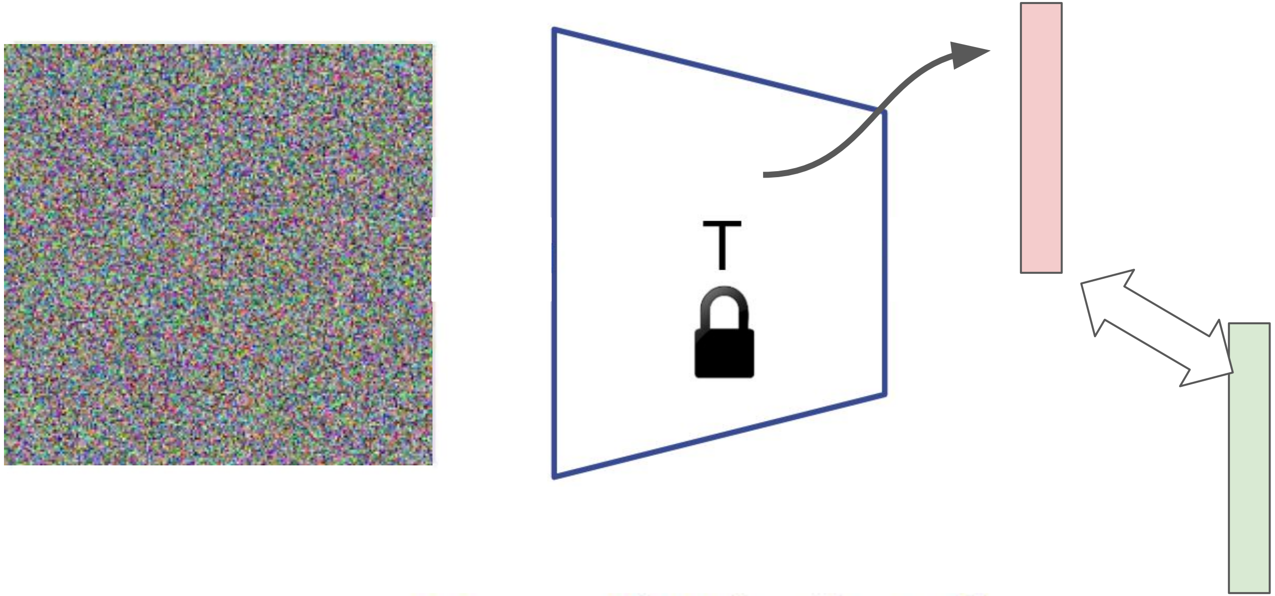
Noise Optimization - Regularizing activation

- Lopes et al. 2018 save the activation summary of the Teacher's layers
  - Mean and Variance

$$\mathcal{R} = l(T(\tilde{x}_i), \phi_i)$$

ϕ_i is the ith layer mean activation saved from the teacher T

Noise Optimization - Regularizing activation



$$\mathcal{R} = l(T(\tilde{x}_i), \phi_i)$$

Noise Optimization - Regularizing activation

- Not data-free
    - Metadata is saved
- Activations can’t represent the complex training data
    - Can’t be applied to sophisticated models

Noise Optimization - Regularizing prediction

- Class and Data Impressions [2018, 2019 & 2021]

$$\mathcal{R} = l(T(\tilde{x}), s)$$

s : is sampled from the softmax space

Noise Optimization - Natural Image Prior

$$\mathcal{R}_{\text{img}}(\tilde{x}) = \mathcal{R}_{\text{TV}}(\tilde{x}) + \mathcal{R}_{l_2}(\tilde{x})$$

Deep Dream [2015] imposes smoothness prior (adjacent pixels to be correlated)

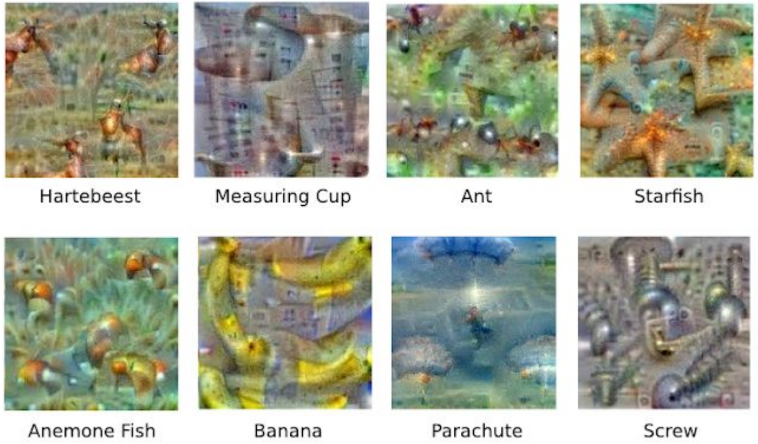
Noise Optimization - Natural Image Prior

$$\mathcal{R}_{\text{img}}(\tilde{x}) = \mathcal{R}_{\text{TV}}(\tilde{x}) + \mathcal{R}_{l_2}(\tilde{x})$$

$$TV(\mathbf{X}) = \sum_{i,j \in \mathcal{N}} ||\mathbf{x}_i - \mathbf{x}_j||_p^q$$

Noise Optimization - Natural Image Prior

$$\mathcal{R}_{\text{img}}(\tilde{x}) = \mathcal{R}_{\text{TV}}(\tilde{x}) + \mathcal{R}_{l_2}(\tilde{x})$$



Deep Dream [2015]

Noise Optimization - BatchNorm Statistics

- BatchNorm layers in CNNs save
    - Running mean and variance → prior about the training data [DeepInversion 2020]

$$\mathcal{R}_{\text{BNS}}(\tilde{x}) = \sum_{l} \left(\|\tilde{\mu}_l(\tilde{x}) - \mu_l\|_2^2 + \|\tilde{\sigma}_l^2(\tilde{x}) - \sigma_l^2\|_2^2 \right)$$

$$\tilde{x}^* = \arg\min_{\tilde{x}} \mathcal{R}_{\text{BNS}}(\tilde{x}) + \mathcal{R}_{\text{img}}(\tilde{x})$$

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Noise Optimization - BatchNorm Statistics

DeepInversion 2020

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Data-driven Intelligence & Learning Lab

Noise Optimization - Summary

- Every sample ← hundreds of gradient ascent/descent steps
  - Computationally expensive
- Quality of the alternate transfer set can't be determined during synthesis
  - Have to perform the KD to evaluate

2

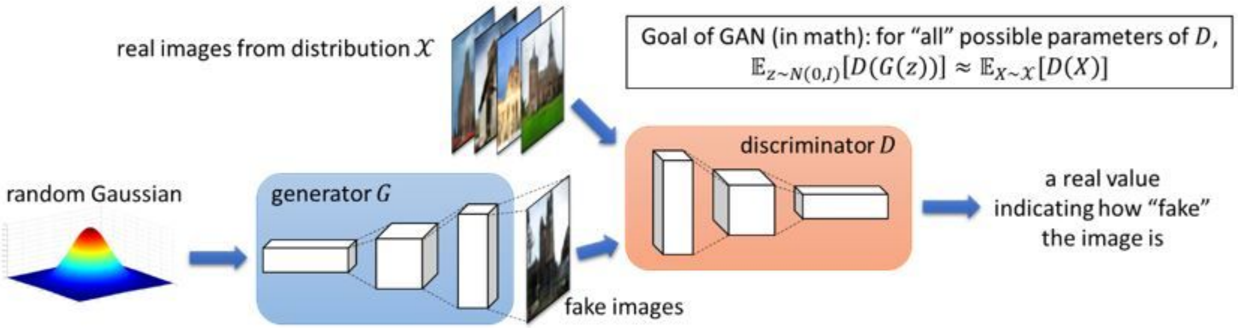
Generative Reconstruction

Adversarial framework for DFKD

Generative Adversarial Network (GAN)

- Generative model to draw high quality samples from the unknown data distribution (ρ_x)
    - Only samples are available from the high-dimensional distribution
- Without computing the densities (ρ_x and ρ_m) it ensures the closeness of the samples

Generative Adversarial Network (GAN)



$$\min_G \max_D \left(\mathbb{E}_{x\sim p_{\text{data}}}[log D(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D(G(z)))] \right)$$

Figure from Microsoft Research Blog

Generative Adversarial Network (GAN)

- G attempts to learn the prior (ρ_x) on the target data with the help of D
  - Adversarial loss navigates G towards ρ_x

GANs for DFKE

- Noise optimization enforces prior about the training data
- Can GANs do it too?
  - Synthesize samples that reflect the training distribution via regularizing
  - Activations
  - Predictions
  - BNS
  - etc.

Generative Reconstruction

Generative Reconstruction

$$\tilde{x} = \mathcal{G}(\boldsymbol{z})\,, \quad \boldsymbol{z} \sim p_z(\boldsymbol{z})$$

$$\underset{G}{\arg\min}\ \mathbb{E}_{z \sim p_z(z)} \mathcal{R}(G(z), T)$$

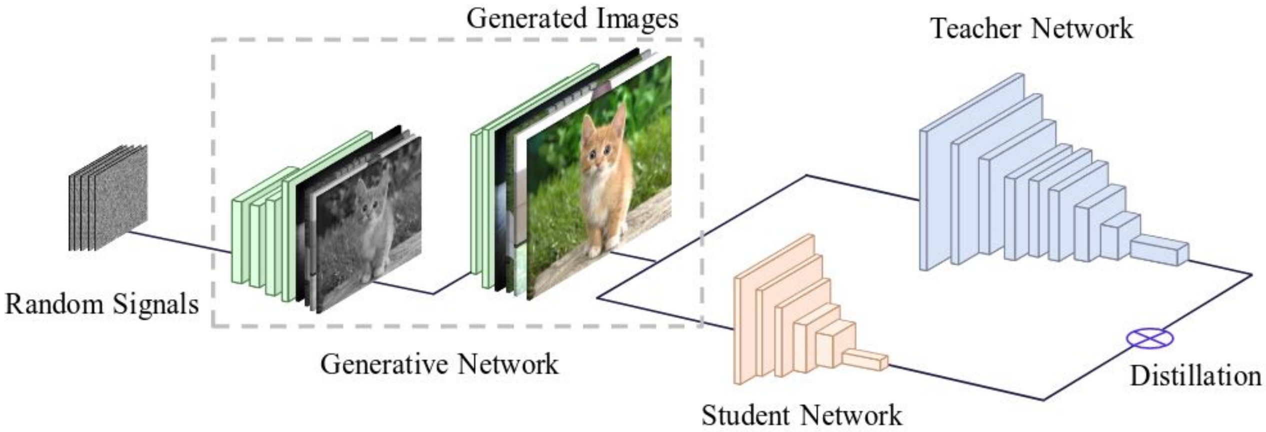
Generative Reconstruction

- Alternate between synthesis and Distillation/Transfer
  - One generator update for each iteration of transfer

Generative Reconstruction

- DAFL 2019 (ICCV 2019) proposed three terms to regularize the generative reconstruction of the training data
- Adapted by later works

Generative Reconstruction



DAFL 2019

$$\mathcal{L}_{oh} = \frac{1}{n}\sum_i \mathcal{H}_{cross}(\mathrm{y}_T^i, \mathrm{t}^i) \qquad \mathcal{L}_a = -\frac{1}{n}\sum_i \|f_T^i\|_1 \qquad \mathcal{L}_{ie} = -\mathcal{H}_{info}(\frac{1}{n}\sum_i \mathrm{y}_T^i)$$

Generative Reconstruction

- DFKA 2021 distill multiple teachers onto a multi-task student
- Luo et al. 2020 Haroush 2020, Besnier 2019 use BNS to distill an ensemble of teachers
- ....

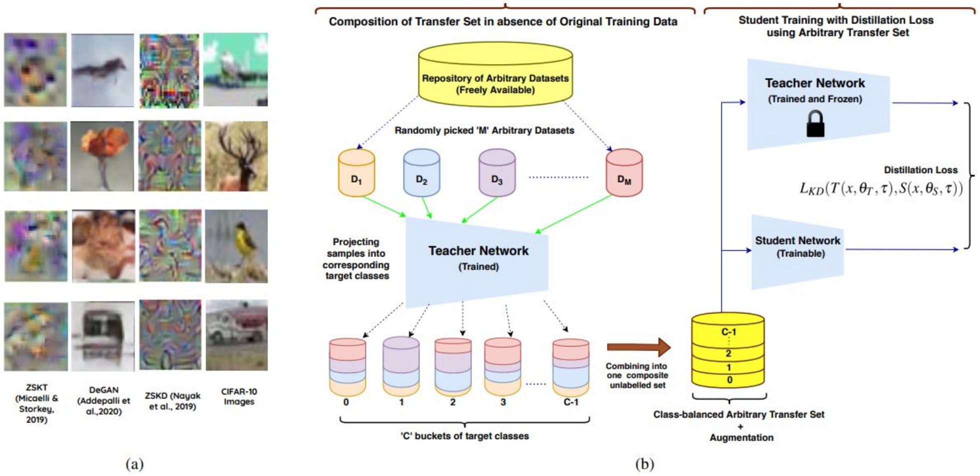
Generative Reconstruction

- DeGAN 2020 employs proxy data as a prior
  - Useful for class incremental learning etc.
- Fang et al. 2021, Han et al. 2021 focus on diversity

Generative Reconstruction

- Nayak & Mopuri et al. 2021 and Chen et al. 2021 explore the arbitrary data as the transfer set
- Non-trivial performance, provided 'class-balancing'

Arbitrary transfer set



[Nayak & Mopuri et al. 2021]

Arbitrary transfer set

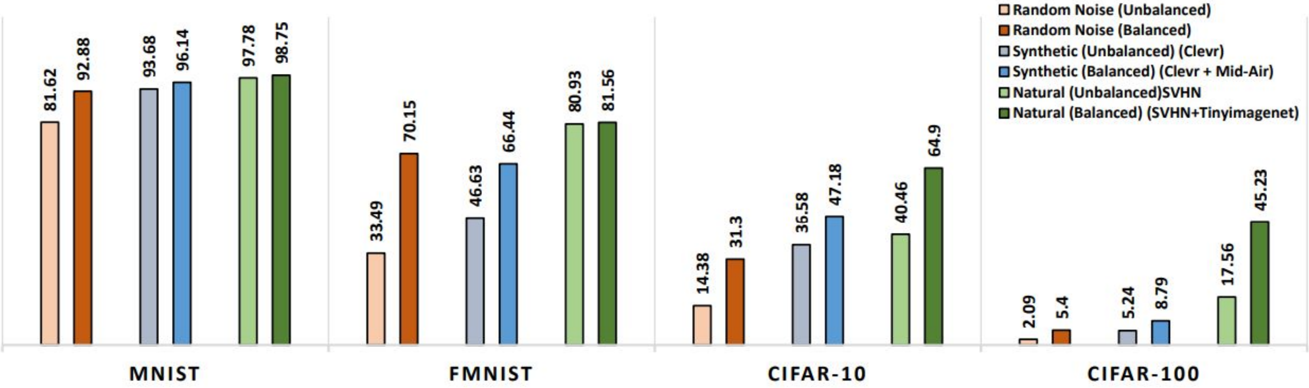


Figure 3. Comparison of the distillation performance using unbalanced and balanced arbitrary transfer sets. Balanced set outperforms its unbalanced counterpart across all the three different varieties of arbitrary datasets: noise, synthetic and unrelated natural data.

Nayak & Mopuri et al. 2021

Weak emphasis on ‘realistic’ reconstructions

- Two approaches so far, aim to reconstruct realistic transfer set
- Still a big gap exists
- Then, why not letting that go and focus more on the transfer?

Focus more on the transfer

- Modification to the Generative reconstruction
    - Make it truly adversarial
    - **T-S** pair penalizes the **G**

Focus more on the transfer

Goal of the DFKD

$$S^* = \arg\min_S \mathcal{D}(T, S)$$

model discrepancy

$$\mathcal{D}(T, S; G) = \mathbb{E}_{z \sim p_z(z)} l(T(G(z)), S(G(z)))$$

Adversarial Exploration

- Goal: Motivate G to generate confusing samples → increase model discrepancy (T vs S)
- Analogous to curriculum learning
  - Progressively challenging samples are presented

But, how to achieve this?

Adversarial Exploration

Update G with -ve discrepancy

$$\arg\min_{G} \; -\mathcal{D}(T, S; G)$$

However, in the distillation phase

$$\arg\min_{G} \; \mathcal{D}(T, S; G)$$

Adversarial Exploration

- Exploration and Transfer/Distillation phases alternate
- G tries to maximize the discrepancy and S tries to minimize (via imitating T)

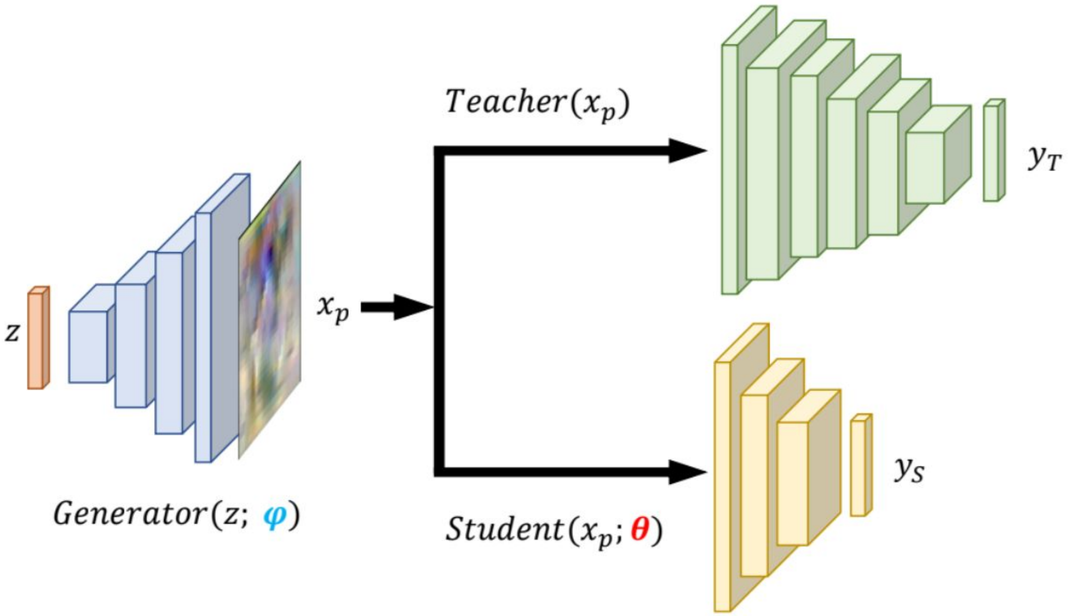
Adversarial Exploration

- Discrepancy can be computed at
  - Feature-based
  - Response-based

Adversarial Exploration

- ZSKT 2019 has successfully demonstrated
  - Call it Adversarial belief matching

Adversarial Belief Matching (ZSKT)



$$\max_{\varphi} \min_{\theta} D_{KL}(y_T, y_S)$$

Figure from Micaelli et al. NeurIPS 2019

Adversarial Belief Matching (ZSKT)

- G searches for the samples on which the T and S disagree
- Then S learns to match T on them
- Adversarial framework makes G to keep exploring the input space

Figure from Micaelli et al. NeurIPS 2019

Generated Images (ZSKT)

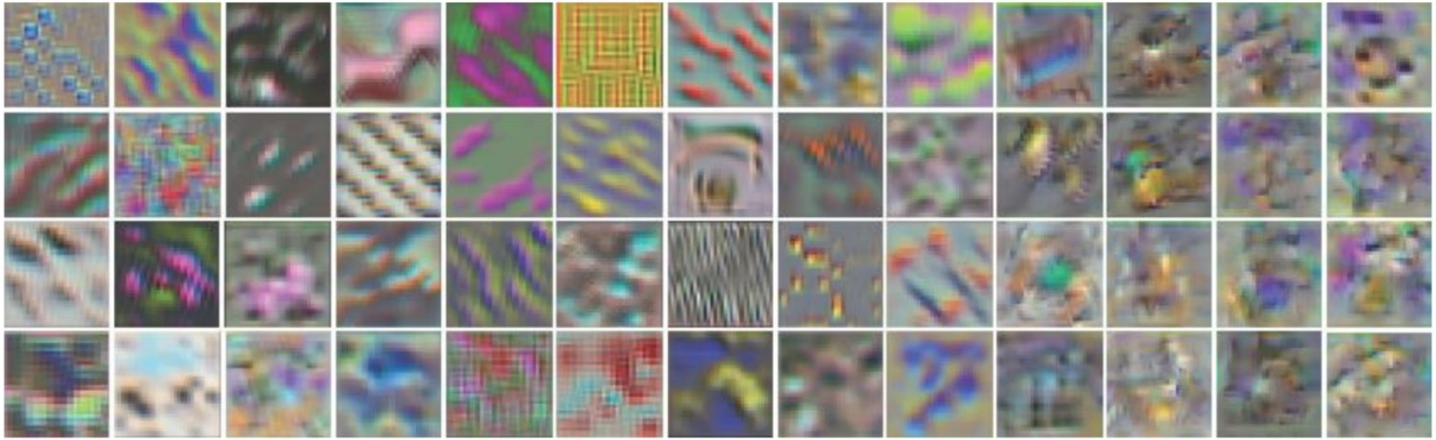


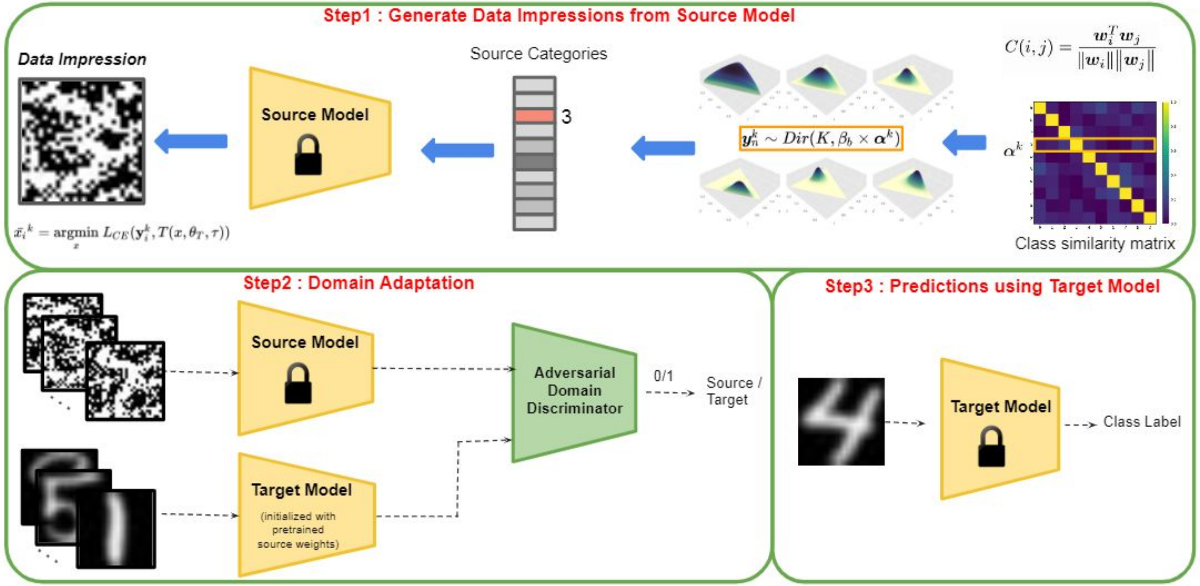
Figure from Micaelli et al. NeurIPS 2019 (CIFAR10)

Adversarial Exploration

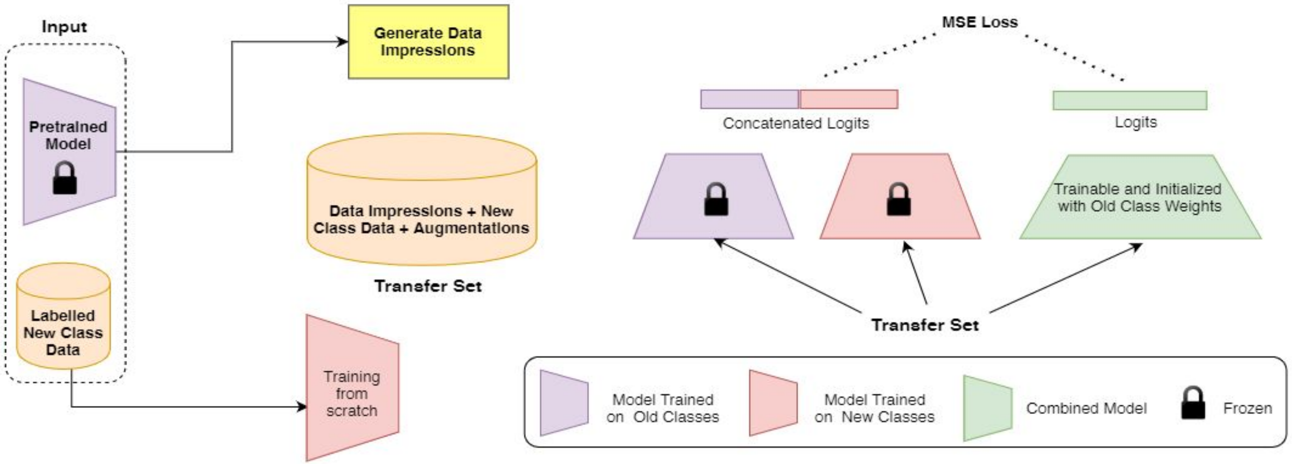
- DFAD 2020
    - investigates for better discrepancy measure
    - extends DFKD to semantic segmentation

Other Applications of DFKE

Domain Adaptation

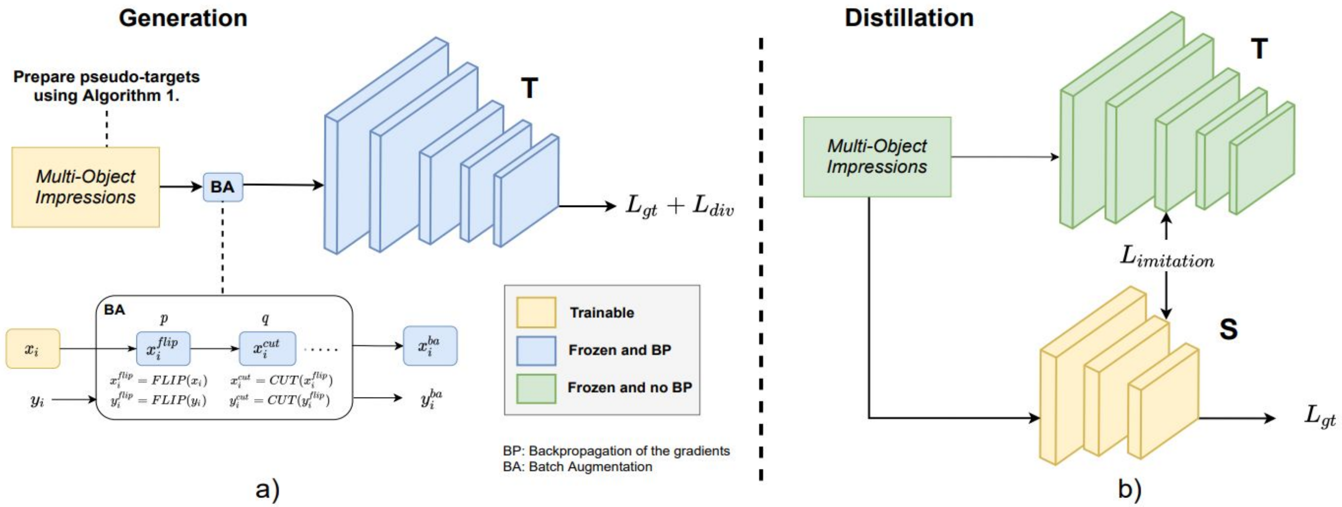


Continual learning



More analysis can be found in Mopuri and Nayak et al. ICML 2019, TPAMI 2021

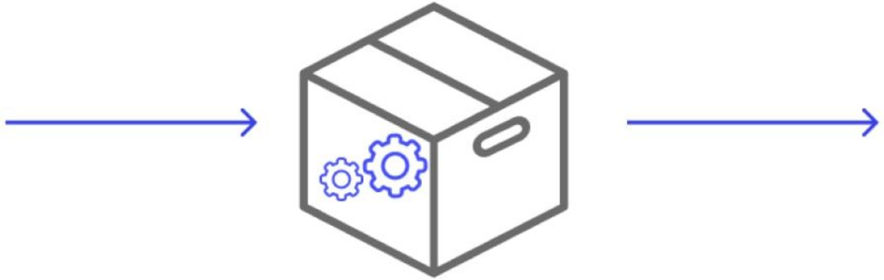
Object detection



Nayak et al [BMVC 2021]

Attributes of full-access (white-box) setting

- Assumes complete access to the model
  - Model architecture/parameters
  - Softmax predictions
  - Gradients



DL Models are Valuable

- Involves data collection and labelling
- Model architecture design and training

DL Models are Valuable

- Involves data collection and labelling
- Model architecture design and training

Models are Intellectual Property (IP) and need protection

Accessing over Cloud (MLaaS)

Machine Learning as a Service (MLaaS)

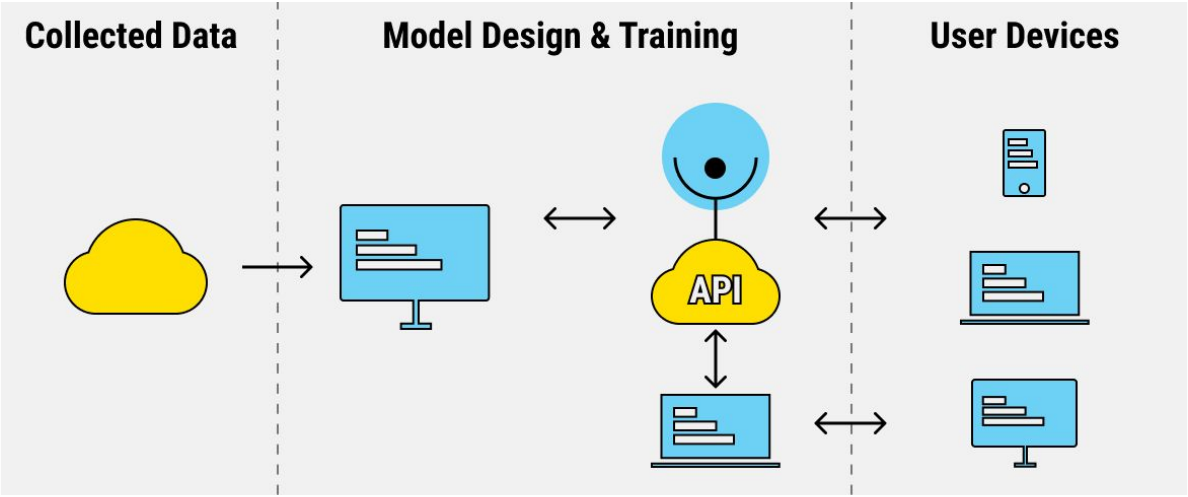
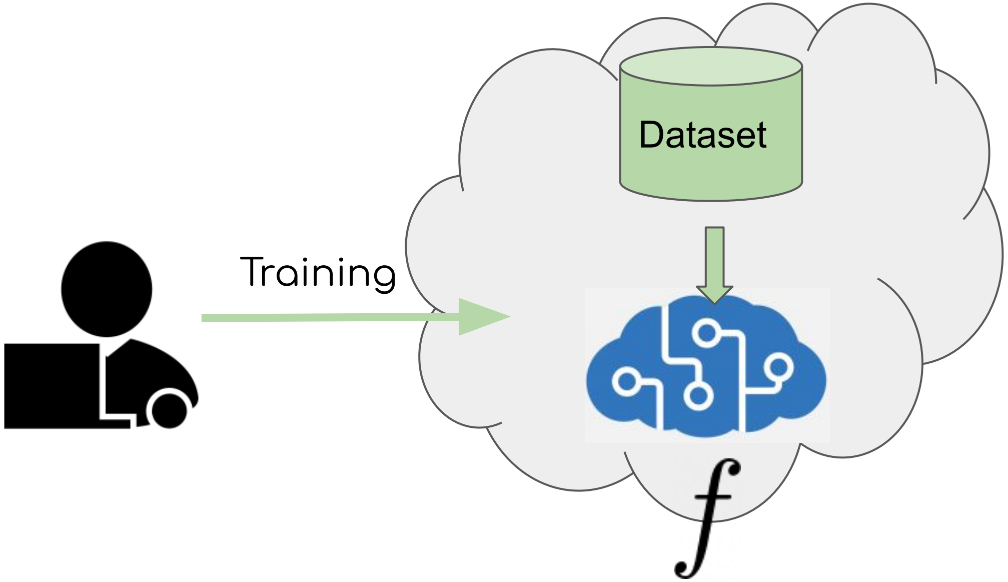
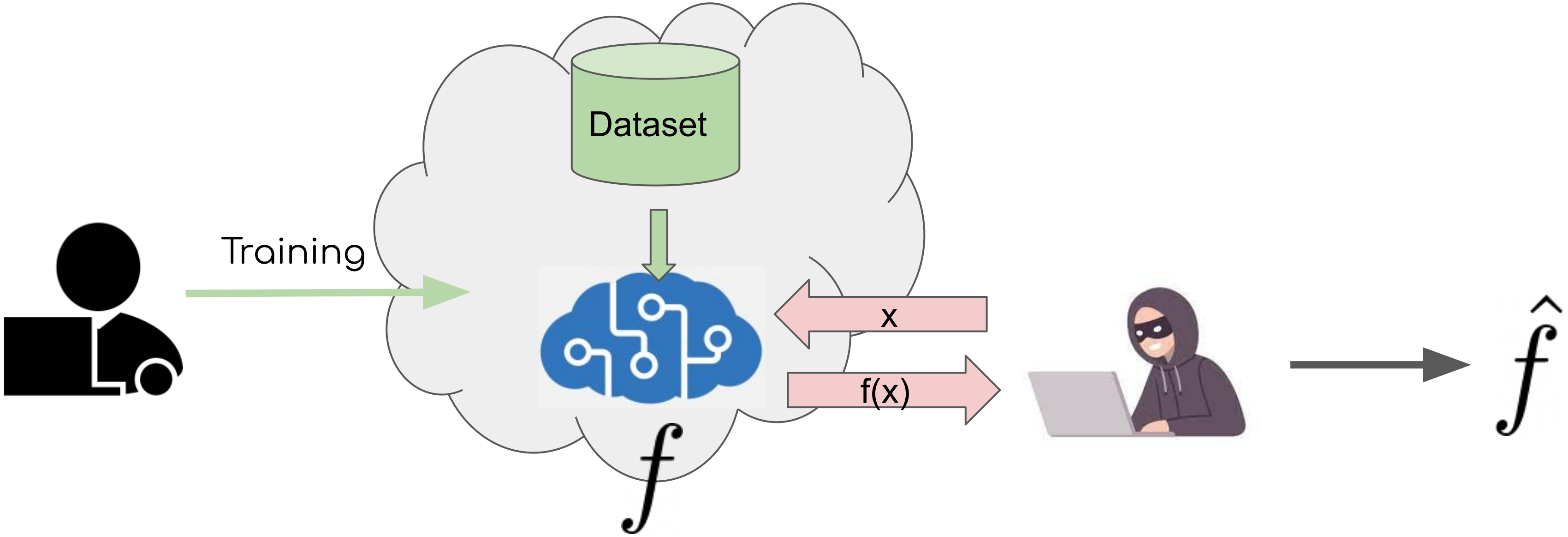


Figure from: https://labelyourdata.com/articles/machine-learning-as-a-service-mlaas

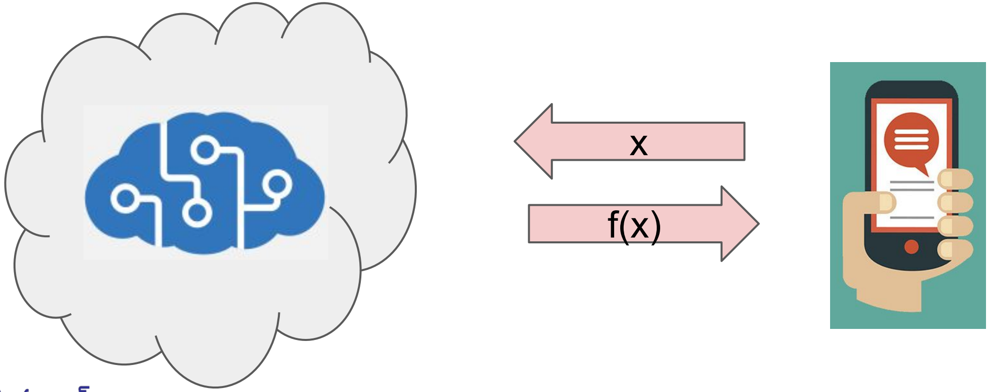
Model Extraction in MLaaS



Model Extraction in MLaaS

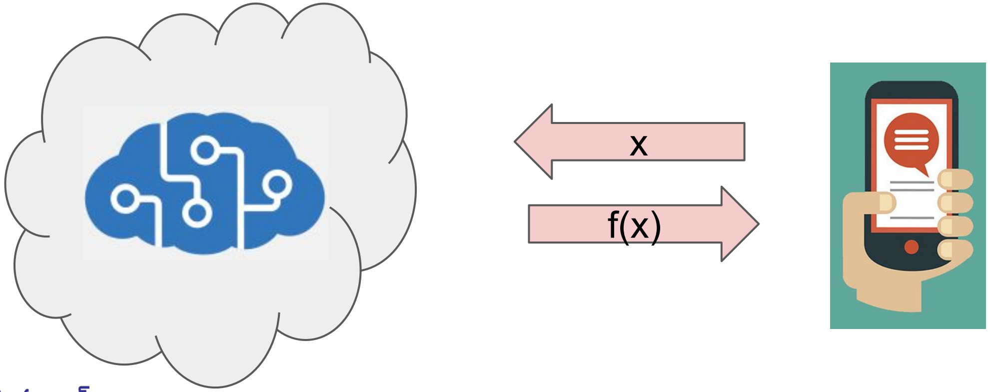


Model Extraction in MLaaS



Model Extraction in MLaaS

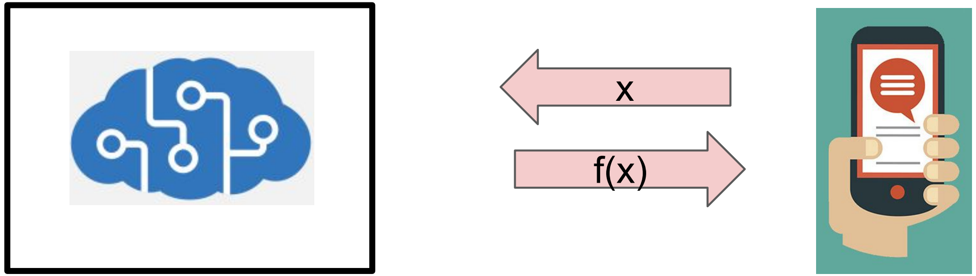
- No access to the model
    - Family, parameters, gradients, softmax, etc.
- One can only generate (x, f(x)) pairs by querying the service



Model Extraction in MLaaS

- No access to the model
  - Family, parameters, gradients, softmax, etc.
- One can only generate (x, f(x)) pairs by querying the service

Black-box setting



Extracting Linear Regression

Extracting a Linear Regression Model

- Hypothesis class: regression model from R^d to R

Extracting a Linear Regression Model

- Hypothesis class: regression model from R^d to R
- A function f in this class can be described with d+1 parameters as

$$f = a_0 + \sum_{1}^{d} a_i x_i$$

Extracting a Linear Regression Model

- An adversary queries $\{x^1, x^2, \ldots x^{d+1}\}$ that are linearly independent
- Can solve the linear system of equations → recover the exact model

$$f = a_0 + \sum_1^d a_i x_i$$

Model Extraction in black-box setting

- Adversarial exploration
- Victim's (Teacher) response only is accessible (no features or gradients)

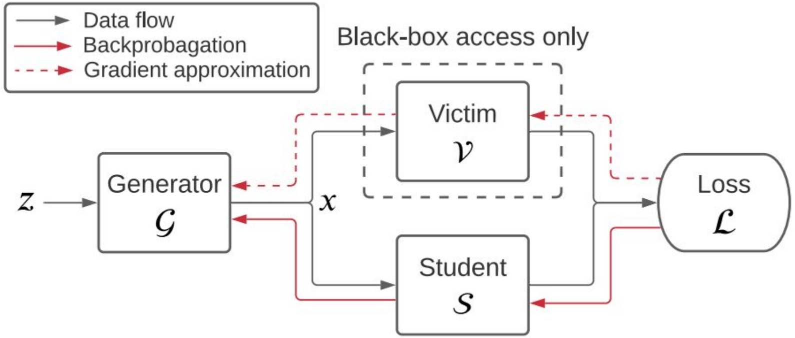


Figure from DFME, CVPR 2021

Model Extraction in black-box setting

Gradient needs to be estimated
e.g. train a surrogate

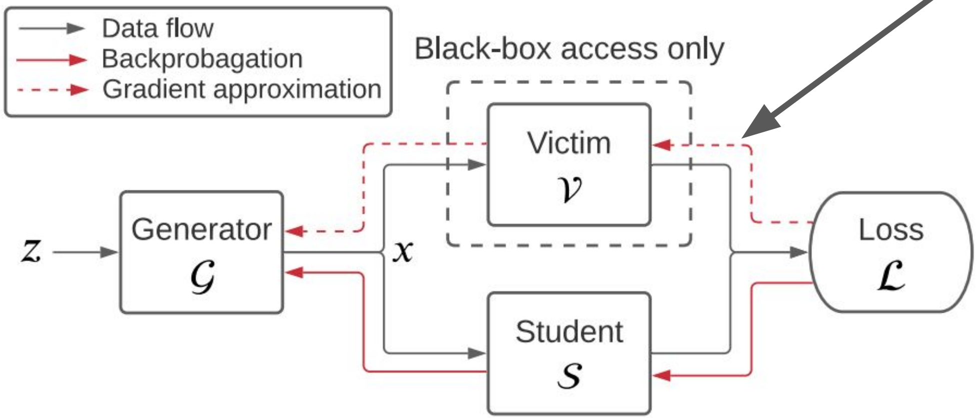
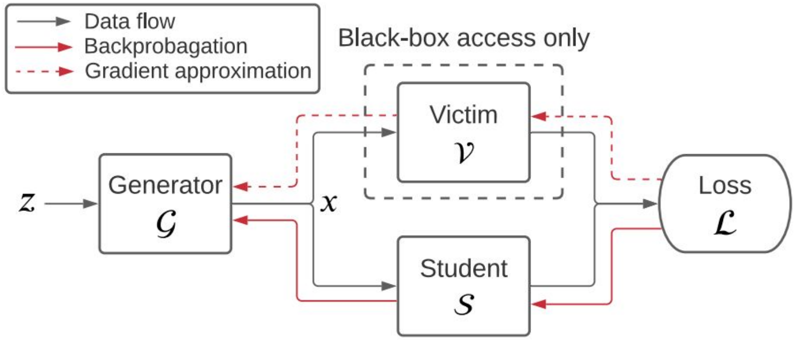


Figure from DFME, CVPR 2021

Model Extraction in black-box setting

- Forward difference method for gradient estimation

$$\nabla_{\mathrm{FWD}} f(x) = \frac{1}{m} \sum_{i=1}^{m} d \frac{f(x + \epsilon \mathbf{u_i}) - f(x)}{\epsilon} \mathbf{u_i}$$



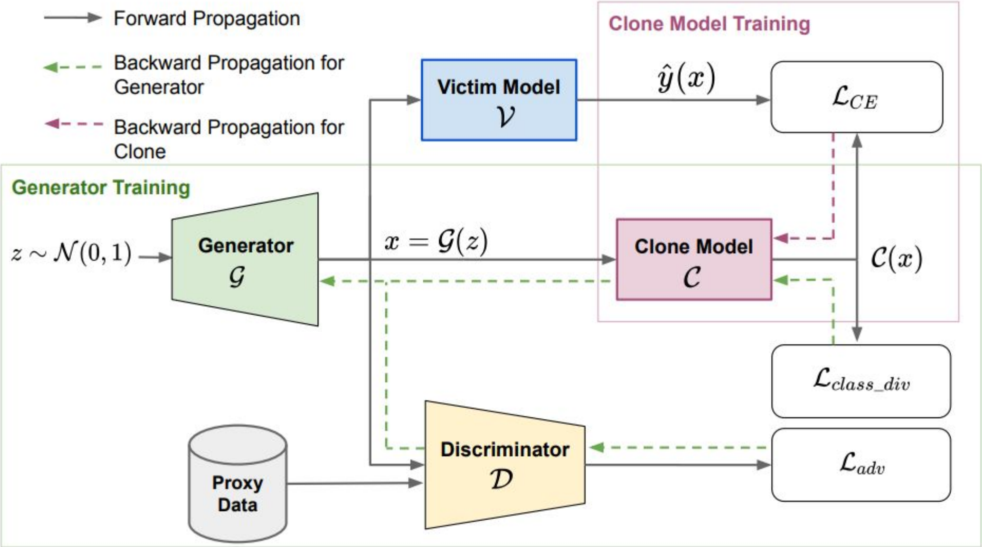
Requires softmax output of the Victim

Figure from [DFME], CVPR 2021

Model Extraction in black-box setting

- Victim models typically give away top-1 label, but not softmax response
- Need to extract the model with 'Hard Label' response

Model Extraction in black-box (or, hard label) setting



G and D training

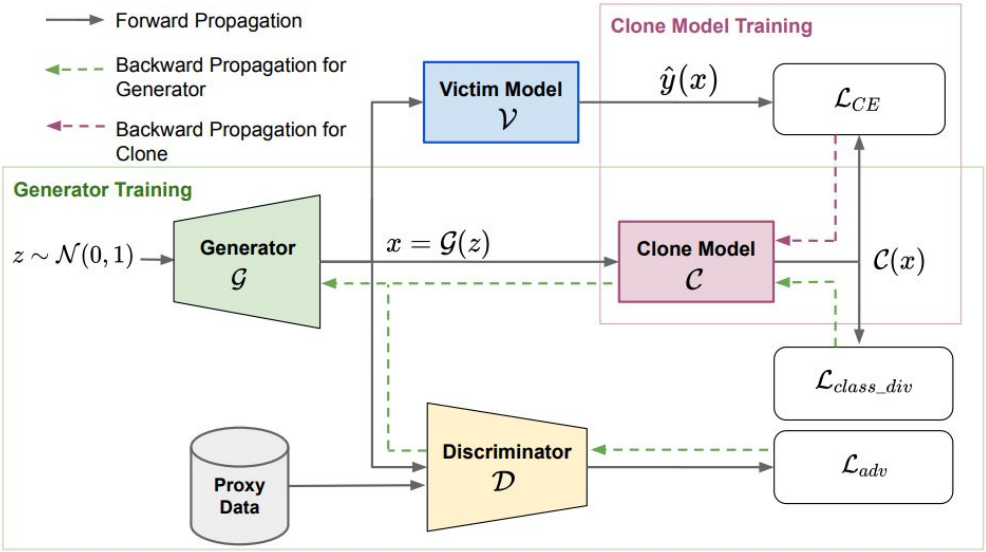
$$\mathcal{L}_{adv,real} = \mathop{\mathbb{E}}_{x \sim p_{data}(x)} [log\mathcal{D}(x)]$$

$$\mathcal{L}_{adv,fake} = \mathop{\mathbb{E}}_{z \sim \mathcal{N}(0,I)} [log(1 - \mathcal{D}(\mathcal{G}(z))]$$

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake}$$

Towards DFME in hard label setting, Sanyal et al. CVPR 2022

Model Extraction in black-box (or, hard label) setting



G and D training

$$\mathcal{L}_{adv,real} = \underset{x \sim p_{data}(x)}{\mathbb{E}} [log \mathcal{D}(x)]$$

$$\mathcal{L}_{adv,fake} = \underset{z \sim \mathcal{N}(0,I)}{\mathbb{E}} [log(1 - \mathcal{D}(\mathcal{G}(z))]$$

$$\underset{\mathcal{G}}{min} \ \underset{\mathcal{D}}{max} \ \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake}$$

$$\mathcal{L}_{class\_div} = \sum_{j=0}^{K} \alpha_j \log \alpha_j$$

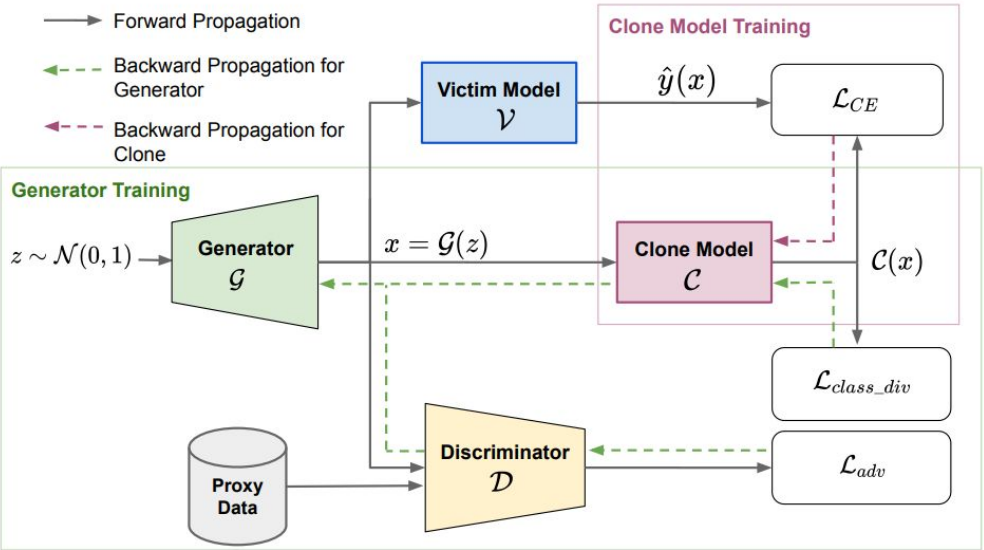
$$\alpha_j = \frac{1}{N} \sum_{i=1}^{N} \text{softmax}(\mathcal{C}(x_i))_j$$

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Towards DFME in hard label setting, Sanyal et al. CVPR 2022

DiL Data-driven Intelligence & Learning Lab

Model Extraction in black-box (or, hard label) setting



G and D training

$$\mathcal{L}_G = \mathcal{L}_{adv,fake} + \lambda_{div}\mathcal{L}_{class\_div}$$

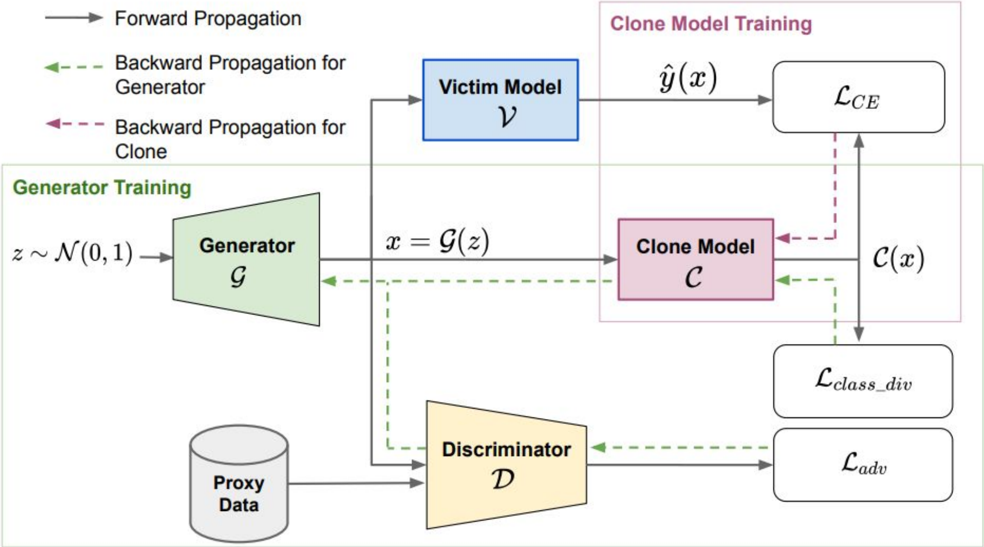
$$\mathcal{L}_D = \mathcal{L}_{adv,real} + \mathcal{L}_{adv,fake}$$

భారతీయ సాంకేతిక విజ్ఞాన సంస్థ హైదరాబాద్
भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Towards DFME in hard label setting, Sanyal et al. CVPR 2022

DiL Data-driven Intelligence & Learning Lab

Model Extraction in black-box (or, hard label) setting



C training

$$\mathcal{L}_C = \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,I)} \left[\mathcal{L}_{CE}(\mathcal{C}(x), \hat{y}(x))\right], \; x = \mathcal{G}(z)$$

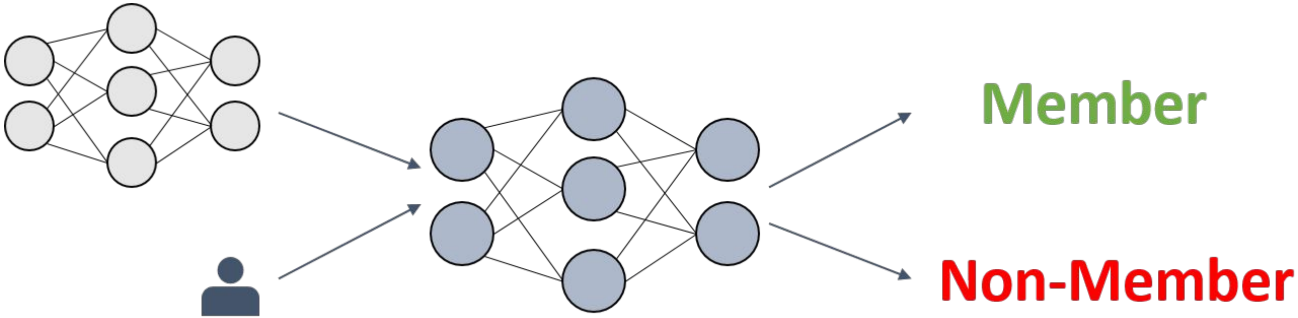
Towards DFME in hard label setting, Sanyal et al. CVPR 2022

Bigger Picture

- Model extraction shares similarities with Active learning
- Dishonest user may launch variety of attacks
  - Membership Inference Attack
  - Model Inversion
  - Model Extraction
  - ....

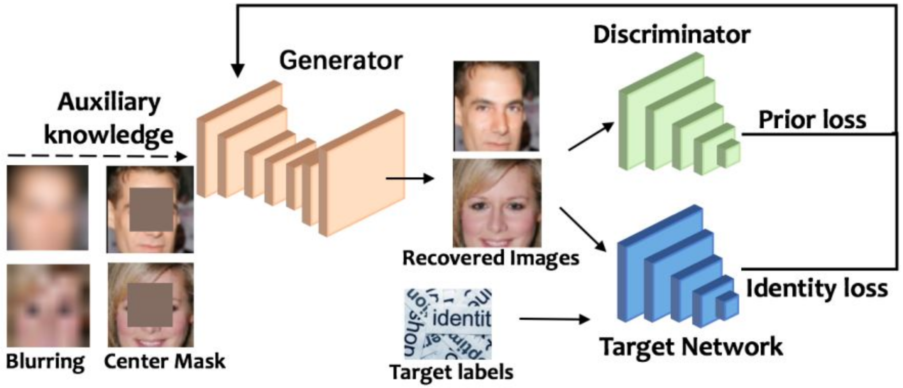
Membership Inference

- Given a data sample (x) and a black-box access to a trained model M, identifying if x was in the training data of [M Shokri et al. 2017]



Model Inversion Attack

- Exploiting the access to a model to infer about a training sample



[GMI Zhang et al. 2020]

Next?

Efficient/Effective Reconstruction

- Quality of the alternate set matters
- Query bandwidth restrictions in MLaaS
- Possibility of 'core' samples identification
    - Via a quick learning loop(?)

Adapting the transfer strategies

- Data-driven transfer strategies may not be ideal for the extracted pseudo samples
    - Customized transfer strategies(?)

Adapting to new scenarios and models

- Distributed/Federated learning
- Sequence models and tasks
- Transformer models
- Generative Models
- Graph neural networks
- ...

Bigger Picture

- Security aspects of ML models needs attention
  - How much of information leakage is possible?
  - Defenses?
- Avenues
  - OOD samples and generalization
  - Prediction power versus the vulnerability

Thank You.