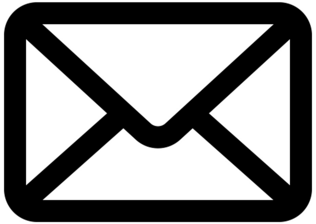
Iterative Teaching by Data Hallucination

Zeju Qiu[1,3]*, Weiyang Liu[1,2]*, Tim Z. Xiao[4], Zhen Liu[5], Umang Bhatt[2,6],
Yucen Luo[1], Adrian Weller[2,6], Bernhard Schölkopf[1]

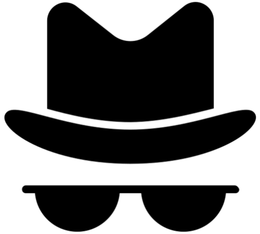
[1]Max Planck Institute for Intelligent Systems, Tübingen, [2]University of Cambridge, [3]Technical University of Munich,
[4]University of Tübingen, [5]Mila, Université de Montréal, [6]The Alan Turing Institute

26th International Conference on Artificial Intelligence and Statistics, AISTATS 2023

1. Introduction



Spam Filter

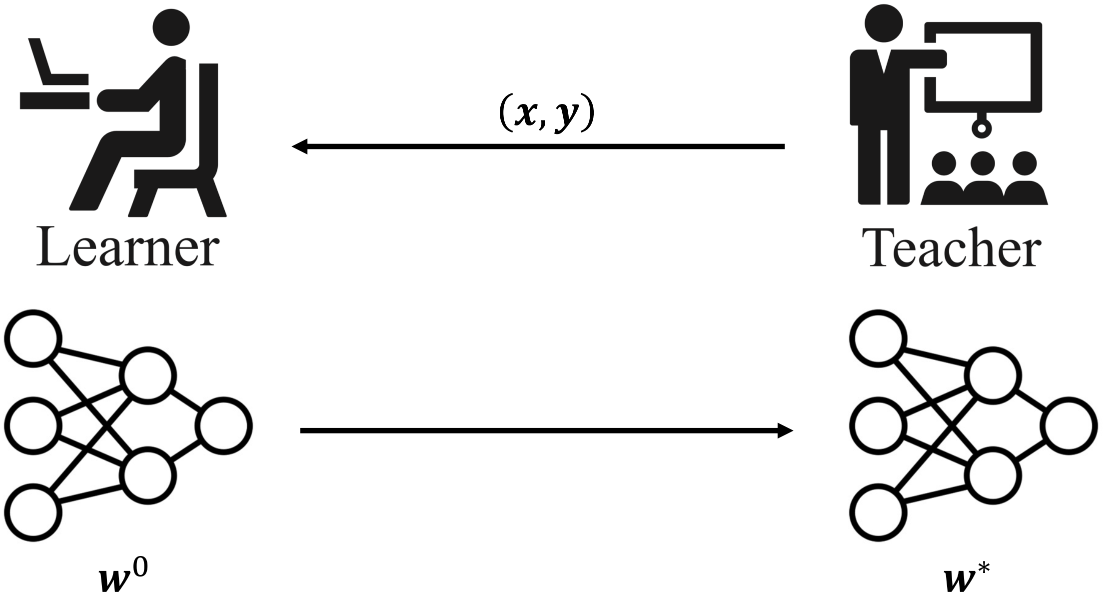


Adversarial Attacker

Training-set poisoning is a **Machine Teaching** problem!

Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), 2016.

1. Introduction



2. Framework

Data Hallucination Teaching Framework (DHT)

- **Idea:** use a generative model / solve an optimization problem to model the teaching process
- **Difference to normal generative model**:
    1. Objective: facilitate convergence (instead of reconstruction)
    2. Models a dynamically changing data distribution depending on the learner's status

Limitations of previous IMT methods:

1. Teacher's limited modeling flexibility / capability ⟶ performance bounded by the data set
2. Inefficient ⟶ goes through the complete data set at each iteration
3. Assume known $\boldsymbol{w}^*$ ⟶ cannot handle black-box teaching of neural networks

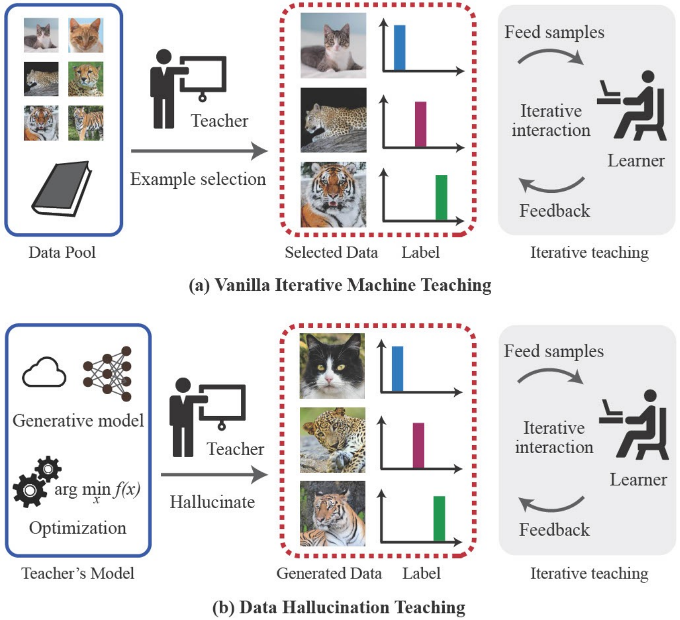


Figure 1: Comparison of vanilla iterative machine teaching and the proposed data hallucination teaching.

3. Problem Setting

Teaching protocol: teacher has access to all the information about the learner (target model parameters $\boldsymbol{w}^*$, model parameters $\boldsymbol{w}_t$, learning rate η, loss function ℓ and optimization algorithm (e.g., SGD)) and can only feed sample pair $(\boldsymbol{x}^i, \boldsymbol{y}^i)$ to the student

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \{\ell(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{w})\}$$

Teacher's objective: the teacher aims to provide examples to the learner in every iteration such that the learner parameters $\boldsymbol{w}$ converge to the desired parameters $\boldsymbol{w}$ as quickly as possible.

$$\min_{\{(\boldsymbol{x}^1,\boldsymbol{y}^1),\cdots,(\boldsymbol{x}^\top,\boldsymbol{y}^\top)\}} d\left(\boldsymbol{w}^T, \boldsymbol{w}^*\right)$$

Learner's objective: the learner minimizes its loss function ℓ with examples given by the teacher.

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \eta_t \frac{\partial \ell\left(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1}|\boldsymbol{w}^t\right)}{\partial \boldsymbol{w}^t}$$

4. Method Overview

Omniscient Data Hallucination Teaching

1. Parametrized Teaching Policy: Data Transformation
2. Parametrized Teaching Policy: Generative Modeling

Black-box Data Hallucination Teaching

1. Mixup-based Teaching
2. Performative Teaching

Main difference between omniscient and black-box machine teaching:

- **Omniscient:** target model parameters $\boldsymbol{w}^*$ **known** $\longrightarrow$ point-to-point distance
- **Black-box:** target model parameters $\boldsymbol{w}^*$ **unknown** $\longrightarrow$ point-to-set distance

4.1 Method: Omniscient DHT

Greedy Teaching Policy

- **Idea**: one-step optimization for the optimal sample $\boldsymbol{x}$ conditioned on a uniformly sampled label $\boldsymbol{y}$
- **More formally:** for each iteration t, solve the teaching problem by optimizing 1 step of $\min_{\{x^{t+1}\}} d(\boldsymbol{w}^{t+1}, \boldsymbol{w}^*)$
- **Formulation:**

$$\min_{\boldsymbol{x}^{t+1} \in \mathcal{X}, \boldsymbol{y}^{t+1} \sim \mathbb{U}} \eta_t^2 \left\| \frac{\partial \ell\left(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1} | \boldsymbol{w}^t\right)}{\partial \boldsymbol{w}^t} \right\|_2^2 - 2\eta_t \left\langle \boldsymbol{w}^t - \boldsymbol{w}^*, \frac{\partial \ell\left(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1} | \boldsymbol{w}^t\right)}{\partial \boldsymbol{w}^t} \right\rangle$$

- **Constraints:** $\boldsymbol{x}$ is within $\mathcal{X}$ (e.g., pixel space $[0, 255]$) and $\boldsymbol{y}$ is within discrete label space
- **Problem:**
  1. Sub-optimal: only consider a one-step update for the learner
  2. Computationally expensive if $\boldsymbol{x}$ high-dimensional (e.g., images)
  3. Not considering data distribution constraints
  4. Not interpretable

4.1 Method: Omniscient DHT

Greedy Teaching Policy

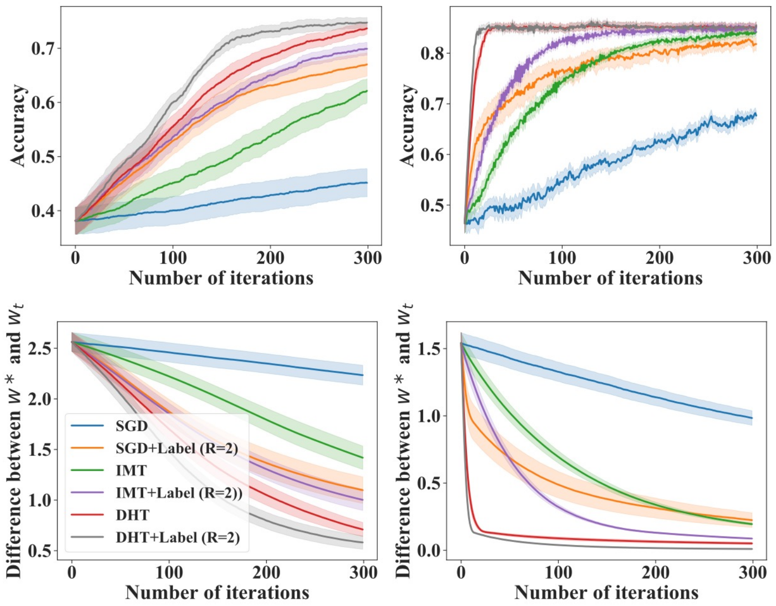


Figure 2: Convergence comparison between our greedy teaching policy with several other baseline methods. Top: half-moon. Bottom: MNIST.

4.1 Method: Omniscient DHT

Parametrized Teaching Policy: Data Transformation

- **Idea**: parametrize the teacher by a neural network $\boldsymbol{\theta}$
- **Teaching policy**: $\pi_{\boldsymbol{\theta}}(\boldsymbol{x}^i, \boldsymbol{y}^i, \boldsymbol{w}_{SG}^i, \boldsymbol{w}^*) = \tilde{\boldsymbol{x}}$
- **More formally:** for each iteration t, solve the teaching problem by optimizing v steps of $\min_{\{x^{t+1},\ldots,x^{t+v}\}} d(\boldsymbol{w}^{t+v}, \boldsymbol{w}^*)$
- **Formulation:**

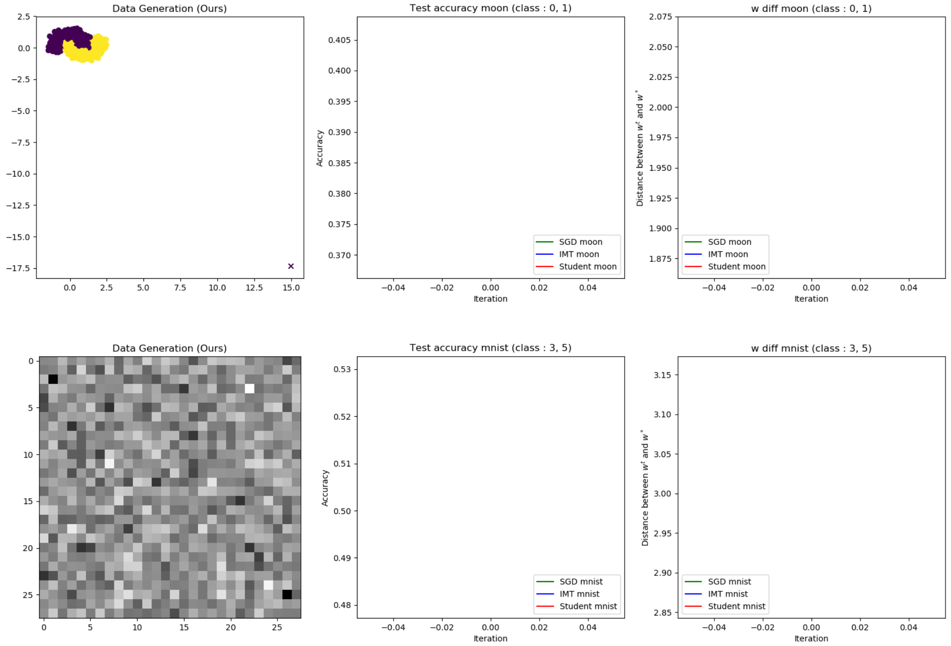
$$\min_{\boldsymbol{\theta}} \|\boldsymbol{w}^v(\boldsymbol{\theta}) - \boldsymbol{w}^*\|_2^2 + \alpha \sum_{i=1}^{v} \ell\left(\pi_{\boldsymbol{\theta}}\left(\boldsymbol{x}^i, \boldsymbol{y}^i, \boldsymbol{w}_{\mathrm{SG}}^i, \boldsymbol{w}^*\right), \boldsymbol{y}^i | \boldsymbol{w}_{\mathrm{SG}}^i\right)$$

$$\text{s.t. } \boldsymbol{w}^v(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \{\ell\left(\pi_{\boldsymbol{\theta}}\left(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{w}^*\right), \boldsymbol{y}|\boldsymbol{w}\right)\}$$

- **Approach**: solve this bi-level optimization by unrolling the inner loop and backpropagate through the whole objective
- **Constraints:** $\boldsymbol{x}$ is within $\mathcal{X}$ (e.g., pixel space $[0, 255]$) and $\boldsymbol{y}$ is within discrete label space
- **Problem:**
  1. Not considering data distribution constraints
  2. Not interpretable

4.1 Method: Omniscient DHT

Parametrized Teaching Policy: Data Transformation



4.1 Method: Omniscient DHT

Parametrized Teaching Policy: Generative Modelling

- **Idea**: parametrize $\pi_{\boldsymbol{\theta}}$ with a generative model and impose a distribution divergence constraint $\mathrm{Div}\big(p(\pi),\ p(\boldsymbol{x})\big) \leq \epsilon$
- **Formulation 1 (GAN):**
  - **Teaching space:** $\pi_{\boldsymbol{\theta}} = \boldsymbol{x}$
  - Teacher $\boldsymbol{\theta}$ act as a generator G

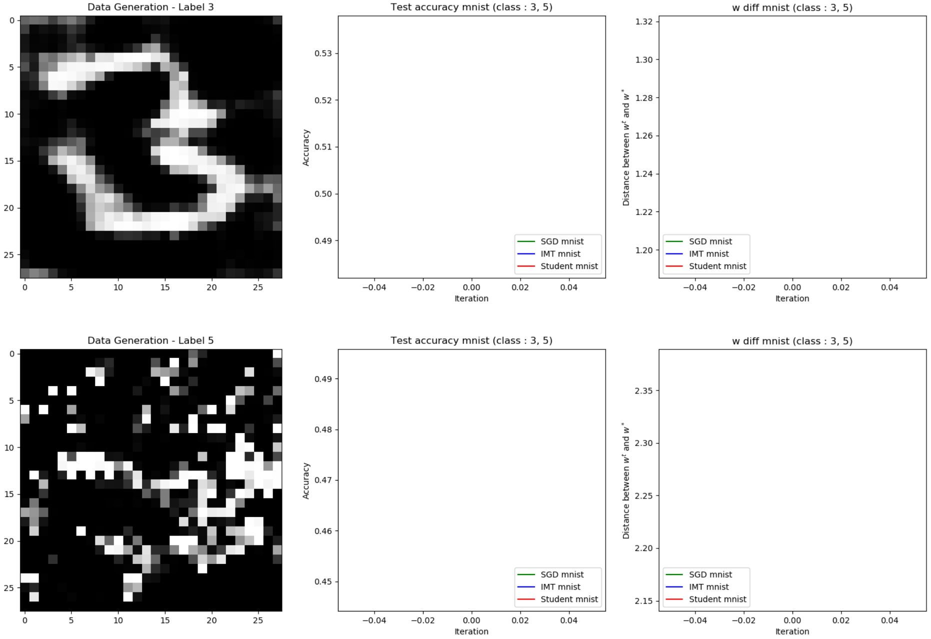
$$\min_{\boldsymbol{\theta}} \max_{D} \mathbb{E}_{\tilde{\boldsymbol{x}} \sim p_{\pi}(\boldsymbol{z})} \log\big(1 - D(\pi_{\boldsymbol{\theta}}(\boldsymbol{z}))\big) + \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x})} \log\big(D(\boldsymbol{x})\big) + \|\boldsymbol{w}^v(\boldsymbol{\theta}) - \boldsymbol{w}^*\|_2^2$$
$$+ \alpha \sum_{i=1}^{v} \ell\big(\pi_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}^i, \boldsymbol{y}^i, \boldsymbol{w}_{\mathrm{SG}}^i, \boldsymbol{w}^*), \boldsymbol{y}^i | \boldsymbol{w}_{\mathrm{SG}}^i\big)$$
$$\text{s.t. } \boldsymbol{w}^v(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y})} \big\{\ell\big(\pi_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{w}^*), \boldsymbol{y}\big) | \boldsymbol{w}\big\}$$

- **Formulation 2 (VAE):**
  - **Teaching space:** $\pi_{\boldsymbol{\theta}} = \boldsymbol{u}$
  - Pre-train VAE on the full data set to capture $p(\boldsymbol{x})$
  - Data recovered using the pre-trained decoder $p_{\varphi}(\boldsymbol{x}, \boldsymbol{y} | \boldsymbol{u})$

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{w}^v - \boldsymbol{w}^*\|_2^2 + \alpha \sum_{i=1}^{v} \ell\left(p_{\psi}\left(\pi_{\boldsymbol{\theta}}\right) | \boldsymbol{w}_{\mathrm{SG}}^i\right) + \mathrm{KL}\left(\pi_{\boldsymbol{\theta}} || p(\boldsymbol{u})\right)$$
$$\text{s.t. } \boldsymbol{w}^v(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y})} \left\{\ell\left(\pi_{\boldsymbol{\theta}}\left(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{w}^*\right), \boldsymbol{y}\right) | \boldsymbol{w}\right\}$$

4.1 Method: Omniscient DHT

Parametrized Teaching Policy: Generative Modelling



4.1 Method: Omniscient DHT

Privacy-preserving Teaching via constrained DHT

- **Motivation**: generating samples that are semantically distinct from the original data distribution could be beneficial
- **Examples**: medical domain, do not wish to leak sensitive information about the patient
- **Goal**: generate samples that differentiate from a privacy set by some distance metrics (e.g., perceptual loss)
- **Formulation**:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{w}^v(\boldsymbol{\theta}) - \boldsymbol{w}^*\|_2^2 + \sum_{t=1}^{v} \ell(\boldsymbol{\pi}_{\boldsymbol{\theta}}, \boldsymbol{y})|\boldsymbol{w}^t) + \max\left\{0, \epsilon - \|\phi(\boldsymbol{\pi}_{\boldsymbol{\theta}}) - \phi(\boldsymbol{x})\|_2^2\right\}$$

$$\text{s.t. } \|\phi(\boldsymbol{\pi}_{\boldsymbol{\theta}}) - \phi(\boldsymbol{x})\|_2^2 \le \epsilon$$

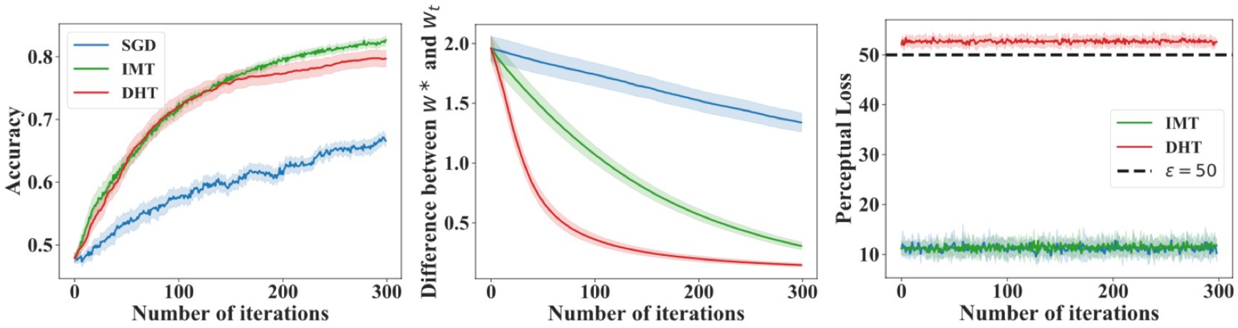


Figure 3: Convergence of privacy-preserving teaching on MNIST. Private perceptual distance of the synthesized samples during teaching. ε is a prescribed distance threshold.

4.2 Method: Black-box DHT

- **Black-box teaching for neural network has been an open challenge in iterative machine teaching!**
- **Goal**: improve the learner's generalization instead of its convergence to some $\boldsymbol{w}^*$
- **Challenges:**
  1. The optimal learner parameters $\boldsymbol{w}^*$ is no longer given
  2. Unclear weight encoding
  3. Difficult to generate plausible samples given no information to $\boldsymbol{w}_t$ and $\boldsymbol{w}^*$

- **Approach:** seek a surrogate for $\boldsymbol{w}^*$ from the underlying joint data and label distribution $\mathbb{P}_{\text{real}} \longrightarrow$ validation accuracy

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbb{P}_{\text{real}}} \{\ell(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{w})\}$$

4.2 Method: Black-box DHT

Mixup-based Teaching

- **Intuition**: teaching space reduction!
- **Idea:** simplify the problem by learning to predict the learner-conditioned interpolation coefficient λ from Mixup
- **Approach**: inspiration using optimization techniques from neural architectural search

- **Surrogate target**: the validation accuracy on a held-out validation data set

- **Teaching space:** $$\pi_{\theta}\big((\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \boldsymbol{w}^t\big) = \lambda_{\theta}\boldsymbol{x}_1 + (1 - \lambda_{\theta})\boldsymbol{x}_2$$

- **Teaching policy:** $$\lambda_{\theta} = h_{\theta}((\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \boldsymbol{w}^t)$$

- **Weight $\boldsymbol{w}^t$**, approximated through model queries:
  - current iteration
  - average training loss
  - best validation loss

Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." *arXiv preprint arXiv:1710.09412* (2017).

4.2 Method: Black-box DHT

Performative Teaching

- **Performative Prediction**: model prediction influences the data distribution
- **Idea**: turn teaching neural network into teaching deep latent features (linear)
- **Approach**:
  - $\boldsymbol{w}^*(t)$ is obtained by updating the linear classifier $\boldsymbol{w}^t$ for v steps
  - $\tilde{\boldsymbol{x}}$ is obtained by feature space perturbation / optimizing along the feature space hypersphere surface
  - ε-neighborhood constraint: $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| \leq \epsilon$
- **Formulation**:

$$\min_{(\boldsymbol{x}_t, \boldsymbol{y}_t)} d\big(\boldsymbol{w}^t, \boldsymbol{w}^*(t)\big)$$
$$\text{s.t. } \boldsymbol{w}^*(t) \sim \mathcal{M}(\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1})$$

- **Intuition**: lookahead optimization, implicit data augmentation
- **Why performative?**

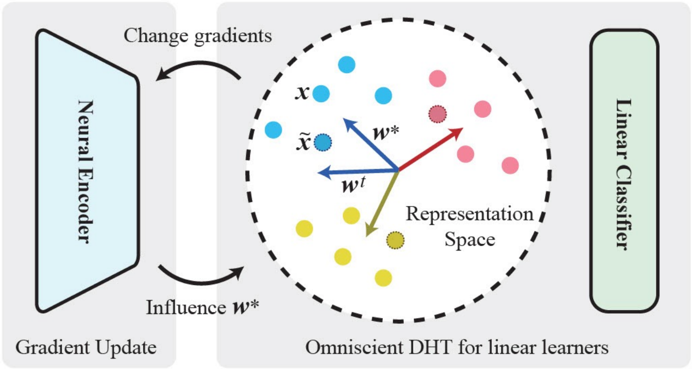


Figure 4: Performative teaching for black-box neural learners.

Wang, Yulin, et al. "Implicit semantic data augmentation for deep networks." *NeurIPS* (2019).
Perdomo, Juan, et al. "Performative prediction." *ICML*. PMLR, 2020.

4.2 Method: Black-box DHT

Performative Teaching

- **Performative Prediction**: model prediction influences the data distribution
- **Idea**: turn teaching neural network into teaching deep latent features (linear)
- **Approach**:
  - $\boldsymbol{w}^*(t)$ is obtained by updating the linear classifier $\boldsymbol{w}^t$ for v steps
  - $\tilde{\boldsymbol{x}}$ is obtained by feature space perturbation / optimizing along the feature space hypersphere surface
  - ε-neighborhood constraint: $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| \leq \epsilon$
- **Formulation**:

$$\min_{(\boldsymbol{x}_t, \boldsymbol{y}_t)} d\big(\boldsymbol{w}^t, \boldsymbol{w}^*(t)\big)$$
$$\text{s.t. } \boldsymbol{w}^*(t) \sim \mathcal{M}(\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1})$$

- **Intuition**: lookahead optimization, implicit data augmentation
- **Why performative?**

Algorithm 1 Performative teaching for neural learners

1. Randomly initialize the neural network. We denote the neural weights of g_1 as $\boldsymbol{v}$, and the neural weights of g_2 as $\boldsymbol{w}$;

for $i = 1, 2, \cdots, T_1$ **do**

2. Form a mini-batch of m samples and perform inference to extract features, denoted as $(\boldsymbol{x}_1^i, \boldsymbol{y}_1^i), \cdots, (\boldsymbol{x}_m^i, \boldsymbol{y}_m^i)$.

3. $\boldsymbol{w}_{\text{buffer}} \leftarrow \boldsymbol{w}$.

4. Fix $\boldsymbol{v}$ and update $\boldsymbol{w}$ by minimizing the empirical risk on the training set (*e.g.*, a few SGD steps).

5. $\boldsymbol{w}^* \leftarrow \boldsymbol{w}$ and then $\boldsymbol{w} \leftarrow \boldsymbol{w}_{\text{buffer}}$.

for $j = 1, 2, \cdots, m$ **do**

6. Solve the greedy teaching problem for the j-th sample:

$$\tilde{\boldsymbol{x}}_j^i = \arg\min_{\boldsymbol{x}} \eta_t^2 \left\| \frac{\partial \ell(\boldsymbol{x}, \boldsymbol{y}_j^i | \boldsymbol{w})}{\partial \boldsymbol{w}} \right\|_2^2 - 2\eta_t \langle \boldsymbol{w} - \boldsymbol{w}^*, \frac{\partial \ell(\boldsymbol{x}, \boldsymbol{y}_j^i | \boldsymbol{w})}{\partial \boldsymbol{w}} \rangle \quad (9)$$
$$\text{s.t.} \left\| \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} - \frac{\boldsymbol{x}_j^i}{\|\boldsymbol{x}_j^i\|} \right\| \leq \epsilon, \quad \|\boldsymbol{x}\| = \|\boldsymbol{x}_j^i\|$$

end

7. Use SGD to update the neural network ($\boldsymbol{w}$ and $\boldsymbol{v}$) by replacing $(\boldsymbol{x}_1^i, \boldsymbol{y}_1^i), \cdots, (\boldsymbol{x}_m^i, \boldsymbol{y}_m^i)$ with $(\tilde{\boldsymbol{x}}_1^i, \boldsymbol{y}_1^i), \cdots, (\tilde{\boldsymbol{x}}_m^i, \boldsymbol{y}_m^i)$.

end

Wang, Yulin, et al. "Implicit semantic data augmentation for deep networks." *NeurIPS* (2019).
Perdomo, Juan, et al. "Performative prediction." *ICML*. PMLR, 2020.

4.2 Method: Black-box DHT

Parametrized Teaching Policy: Generative Modelling

Dataset	Learner	SGD	Random Policy	DHT
MNIST	MLP	92.45 ± 0.07	92.47 ± 0.06	$\mathbf{95.02 \pm 0.04}$
CIFAR-10	CNN-3	87.30 ± 0.28	87.17 ± 0.17	$\mathbf{88.77 \pm 0.35}$
	CNN-6	90.34 ± 0.10	90.20 ± 0.09	$\mathbf{91.61 \pm 0.23}$
	CNN-9	91.10 ± 0.26	91.12 ± 0.12	$\mathbf{92.30 \pm 0.13}$
	CNN-15	91.85 ± 0.28	91.67 ± 0.13	$\mathbf{92.44 \pm 0.15}$
CIFAR-100	CNN-3	62.10 ± 0.29	62.04 ± 0.11	$\mathbf{62.69 \pm 0.37}$
	CNN-6	65.02 ± 0.24	64.96 ± 0.17	$\mathbf{66.81 \pm 0.17}$
	CNN-9	67.05 ± 0.29	67.19 ± 0.23	$\mathbf{69.23 \pm 0.34}$
	CNN-15	68.39 ± 0.39	68.49 ± 0.17	$\mathbf{68.96 \pm 0.36}$

Table 1: Testing accuracy (%) of performative teaching. Multiple types of neural learners (e.g., MLP and CNN) are considered.

Method	Accuracy (%)
ERM	61.75
cMixup	66.27
dMixup	65.80
Unrolling	65.18
Policy gradient	**67.64**

Table 2: Empirical results on CIFAR-10.

5. Summary

Contribution

- We propose a novel teaching framework Data Hallucination Teaching (DHT), where the teacher iteratively generates synthetic training data depending on the learner's status. DHT yields a highly flexible teaching space.
- In the DHT framework, we comprehensively study the teaching policies under both the omniscient and black-box scenarios.
- We propose a novel performative formulation for iterative teaching, which assumes a dynamically changing teaching target. The formulation is shown to be a natural fit for teaching black-box neural learners.
- For the first time, we are able to apply iterative teaching to black-box neural learners on realistic datasets. Significant performance gain is observed empirically.
- We demonstrate faster convergence of DHT versus SGD and other baselines, both theoretically and empirically.