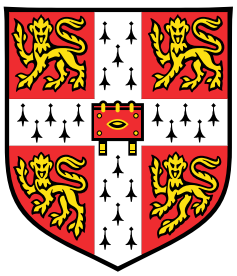
# Establishing a Unified Framework for Iterative Machine Teaching



**Muqing Xue**

Department of Engineering
University of Cambridge

This thesis is submitted for the degree of
*Master of Philosophy in Machine Learning and Machine Intelligence*

Downing College August 2023

To my parents.

You are the cornerstone of my every journey.

# Declaration

I, Muqing Xue of Downing College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

This dissertation contains 11815 words excluding declarations, bibliography, photographs and diagrams but including tables, footnotes, figure captions, appendices and abstract.

**Software Declaration** All experiments in this project are implemented in Python, utilising standard machine learning applications such as PyTorch. All codes are developed by myself, with reference to the public code base of (Qiu et al., 2022), and portions of code for (Liu et al., 2021) provided by Weiyang Liu, the co-supervisor of this project.

Muqing Xue
August 2023

# Acknowledgements

This thesis concludes my very intense and vibrant year at Cambridge. Amidst the relentless new challenges and experiences, I have learned so much in this one year's time.

I would like to start by thanking Dr. Adrian Weller, who has introduced me to the topic of machine teaching and its potential connection with game theory, and Weiyang Liu, my co-supervisor and the proposer of iterative machine learning. Despite our physical distance, thank you for the time and effort you dedicated to supervising me. Our discussions, especially during moments when I felt lost most, have been truly helpful. It has been my pleasure to work with both of you. I extend my gratitude to Zeju Qiu, the co-author of DHT, for addressing my questions about the paper, particularly regarding the intricate implementation details.

I must also thank Dr. John Dudley, the incredible MLMI course director who has patiently addressed our myriad questions, and who has also graciously served as my course advisor. Equally, a heartfelt thanks is due to Dr. Marta Correia. Thank you both for your unwavering kindness and support throughout the year, especially during the more challenging times.

Thank you to the MLMI cohort for a rich and joyful year. In particular, to Jiajun He, for the countless help and our many intriguing and inspiring conversations, which have sparked many ideas for the upcoming sketch.

A very special thank you to Yuncheng Lu, for the GPU and TikZ genius, the accompany and support, and the positivity and faith.

The last and foremost, thank you to my parents and family. Distanced by times and miles, now and then, you are the cornerstone of my every journey.

*To my year of Master of Philosophy in Machine Learning and Machine Intelligence at the University of Cambridge.*

Once when my father drove me home from middle school, I asked him what is philosophy. He said: “Should you study any subject to depth, it is philosophy.”

People dream that machines can learn like human, but perhaps we also live and learn like machines. On-policy.

The process is stochastic, incurring rewards or losses in real-time. Rational players live in the present and look ahead, maximising the expected marginal payoff. While the current displacement may appear like today, the higher derivatives that lie in spirit drive the trajectory into the future.

At the harder times, the higher losses may hurt the most, but with larger gradients that is often when we learn the fastest, sometimes escaping local optima and developing new modes. Overcoming obstacles, time and again, we proudly evolve into mature, robust individuals with multi-model distribution.

The great simulator of the world does not reset or turn back in time, we thus cannot restart or reinitialise, or have multiple attempts. We all live one and only one life, from different initialisation, after different objective functions, and at different learning rates. We seem to be disadvantaged compared to machines in optimisation, but are in fact much luckier than any algorithm — in that it is not optimisation what life is about, but the trajectory itself is the meaning.

May you converge, may you converge not.

# Abstract

Machine teaching (MT) studies the problem of providing "optimal" data to guide the training of machine learning models. While MT started by considering the construction of an optimal (usually minimal) dataset, iterative machine teaching (IMT) is proposed in 2017, taking into consideration the iterative nature of modern learning algorithms and providing examples sequentially based on the latest student. In the current machine learning era, IMT has found its various applications and connections with other studies of interest.

In this thesis, we review the key line of work in IMT. By identifying three key dimensions, we propose a unified framework that organises the various methods as a coherent set of ideas. Guided by this framework, we are able to systematically investigate the existing methods, providing further insights in some cases and exposing current limitations in others, Finally, we review IMT and its related studies in a more general view, providing a unified taxonomy of data-wise training aid methods.

# Table of contents

**Nomenclature** **xiii**

**1 Introduction** **1**
1.1 Thesis Contribution . . . . . 3
1.2 Thesis Outline . . . . . 3
1.3 Problem Setting . . . . . 4

**2 The "OG" Case** **7**
2.1 Omniscient Teacher . . . . . 7
2.1.1 Why Care? . . . . . 8
2.2 Greedy Teacher . . . . . 8
2.2.1 Sample Selection . . . . . 9
2.2.2 Label Synthesis Teaching (LAST) . . . . . 10
2.2.3 Data Hallucination Teaching (DHT) . . . . . 12
2.3 Dimension 1: Teaching Capability . . . . . 12
2.4 Experiments: Comparing across Teaching Capability . . . . . 14
2.4.1 Experimental Setup . . . . . 14
2.4.2 Results . . . . . 15
2.5 Summary . . . . . 19

**3 Parametrised Teacher** **21**
3.1 Methodology . . . . . 21
3.2 Experiments: Practical Challenges . . . . . 26
3.3 Summary . . . . . 27

**4 Black-box Teacher** **29**
4.1 What does black-box mean? . . . . . 29
4.2 Case 1: Imitation Teacher . . . . . 30

4.3 Case 2 . . . 30
4.3.1 Surrogate Teacher . . . 30
4.3.2 Performative Teacher . . . 31
4.4 Summary . . . 32

**5 A Unified IMT Framework** **33**
5.1 Characterising the IMT Space . . . 33
5.1.1 Dimension 2: Teaching Scope . . . 34
5.1.2 Dimension 3: Teaching Objective . . . 34
5.2 Unifying beyond IMT . . . 35
5.3 Summary . . . 36

**6 Related Research Areas** **37**
6.1 Taxonomy . . . 37
6.2 Methods Synopsis . . . 39
6.2.1 Training Set Cardinality . . . 39
6.2.2 Convergence Speed . . . 41
6.2.3 Generalisability . . . 42
6.3 Summary . . . 43

**7 Conclusion** **45**
7.1 Open Questions . . . 45

**Appendix A Intuitive Interpretation** **47**

**Appendix B Theoretical Results** **51**
B.1 Overview . . . 51
B.2 Universal Speedup . . . 52
B.2.1 Joint Teacher . . . 52
B.2.2 LAST . . . 53
B.3 Exponential Speedup . . . 53
B.3.1 Joint Teacher . . . 54
B.3.2 LAST . . . 54
B.4 Summary . . . 55

**Appendix C Experimental Details** **57**

**References** **59**

# Nomenclature

**Roman Symbols**

| | |
|---|---|
| $\mathcal{D}$ | Dataset |
| $f$ | Student's model |
| $l$ | Loss function |
| $w$ | Student's parameter |
| $w^*$ | Optimal parameter |
| $x$ | Input |
| $\tilde{x}$ | Synthesised input |
| $y$ | Label |
| $\tilde{y}$ | Synthesised label |

**Greek Symbols**

| | |
|---|---|
| $\eta$ | Learning rate |
| $\pi$ | Teacher's network |
| $\theta$ | Teacher's parameter |

**Acronyms / Abbreviations**

| | |
|---|---|
| DD | Dataset Distillation |

DHT Data Hallucination Teaching

DTA Data-wise Training Aids

IMT Iterative Machine Teaching

L2T Learning to Teach

LAST Label Synthesis Teaching

MT Machine Teaching

SGD Stochastic Gradient Descent

# Chapter 1

# Introduction

**Machine teaching** (MT), previously characterised by the series of work by Zhu (2013, 2015), studies the problem of constructing an optimal (usually minimal) dataset that will guide a student model to learn a target concept. While the origin of its idea can be traced back to the theoretical study of *teaching dimension* in the past century (Goldman and Kearns, 1995; Shinohara and Miyano, 1991), MT gains a surge of interest in the last decade. With a shifted focus to modern machine learning models and generalised objectives beyond training set cardinality, MT finds its diverse applications in education (Xu et al., 2021) and cyber-security (Mei and Zhu, 2015; Alfeld et al., 2016, 2017), as well as close connections with other topics of interest such as curriculum learning (Bengio et al., 2009), knowledge distillation (Hinton et al., 2015) and dataset distillation (Wang et al., 2018).

While this previous line of work focused on the construction of datasets, a crucial reality to recognise is that, nowadays, the training of machine learning models (in particular, neural networks) almost exclusively uses iterative algorithms, typically stochastic gradient descent (SGD) or its variants. Thus, the final model trained depends on not only the model architecture, but also the learning algorithm; and not only on the dataset, but also the order in which data are used by the model in each iteration.

In 2017, Liu et al. first proposed **iterative machine teaching** (IMT), where we take into consideration the iterative nature of model training and consider data to use at each iteration, potentially based on the student's status after the last iteration. Correspondingly, our goal changes from constructing the minimal dataset to achieving the minimal number of iterations, i.e. fast convergence.

Shortly following the introduction of this new paradigm, a succession of works emerged (Liu et al., 2018, 2021; Xu et al., 2021; Qiu et al., 2022; Zhang et al., 2023), expanding the range of methods and broadening the scope of IMT. The teacher's option is no longer restricted to *selecting* samples from a static dataset, but also includes *synthesizing* label or input image in a continuous teaching space. IMT further draws links to soft label training (Hinton et al., 2015; Szegedy et al., 2016), data augmentation (Chen et al., 2020; Krizhevsky et al., 2012) and generative models (Kingma and Welling, 2013; Goodfellow et al., 2020). Moreover, the objective of IMT also extends from focusing on convergence speed to encompassing generalizability (Qiu et al., 2022).

Amidst the rapid expansion of novel methodologies, one naturally wonders: How do these different IMT methods compare – what are their commonalities, and what are their key differences? Moreover, how exactly does IMT connect with such a broad spectrum of machine learning studies, and how do these studies relate to each other?

In my thesis, we review the key line of works in IMT and "perform principle component analysis (PCA)" – By comparing different formulations, we identify three key dimensions that underlie the range of methods and thus provide a unified framework that organises them as a coherent set of ideas.[1] With this framework, we can easily compare and contrast different methods, or identify unexplored gaps in existing works.

Also guided by this framework, we can organise experiments that fairly compare methods along specific dimensions. We perform such experiments along the first dimension for methods in the most well-established case of omniscient, greedy teaching. However, for more ambitious teaching strategies and challenging settings across the second and third dimensions, our experiments have shown strong and non-trivial conditions they require to work in practice, which greatly limits their universal applicability at the current stage. Nonetheless, we discuss the ideas of the proposed strategies.

Finally, we review the variety of studies related to IMT and perform another level of PCA – By comparing IMT and the related studies, we identify their commonality. Namely, they all aim to improve model training in the aspect of data. We further provide a taxonomy that classifies them in a unified manner.

In summary, we hope this work can serve as a land map for IMT. By reviewing the series of IMT works, we provide a succinct summary of the various methods proposed and highlight

[1] This organisation draws inspiration from the review of MT by Zhu et al. (2018). In fact, IMT can further fit into their taxonomy of general sense MT as sequential teaching.

some current limitations. By viewing IMT as a whole along with its related works, we locate IMT's position among the various studies around machine learning. At both level, we hope to provide a clearer view of the field, making connections and distinctions explicit, and possibly revealing untapped potential for future researchers.

## 1.1 Thesis Contribution

The contribution of this thesis includes:

- A systematic and narrative review of the key line of IMT methods (Chapter 2-4), leading to a unified framework characterised by three key dimensions (Chapter 5)
- Empirical examination of IMT methods that uncovers insights not identified in prior works, specifically includes:
  - A systematic comparison of methods of different teaching capabilities in omniscient, greedy teaching (Section 2.4)
  - Experiments revealing several limitations of parameterised teachers, in terms of delicate training, strong dependency on student initialisation and locally optimal performance (Section 3.2)

  Thereby, we also provide a unified code base for the various IMT methods, available at https://github.com/DianaS1108/IMT
- A taxonomy of IMT and its related studies that share the common goal of optimising model training in the aspect of data, which we collectively refer to as data-wise training aids (DTA)

## 1.2 Thesis Outline

The remainder of the thesis is structured as follows.

While the proposed IMT is a general concept, in this work, we mainly consider a classical supervised learning setting, which we first specify in Section 1.3.

In Chapter 2-4, we introduce the key line of methods in IMT, in a refactored order which will build towards a unified framework in Chapter 5. Following the "easy-to-hard" spirit of curriculum learning (Bengio et al., 2009), we start with the most basic case of omniscient,

greedy teaching, and gradually move to the more sophisticated parameterised teaching and generalised teaching objectives. Specifically,

- In Chapter 2, we focus on omniscient, greedy teachers and introduce three teaching methods: the baseline IMT (Liu et al., 2017), LAST (Liu et al., 2021) and DHT (Qiu et al., 2022). By comparing them, we identify the first dimension of our unified IMT framework: teaching capacity. Accordingly, in Section 2.4, we structure a set of experiments to investigate its effect.
- In Chapter 3, we aim to improve upon the suboptimality of the greedy strategy and study parameterised teachers. We discuss their methodology, however, in Section 3.2, our experiments show non-trivial implementation nuances and limitations of these methods in practice.
- In Chapter 4, we touch upon the even more challenging problem of black-box teaching. We first clarify its different meanings in literature, then outline the different strategies that have been proposed.
- In Chapter 5, we review and reflect on the IMT methods introduced, identify the remaining two dimensions and present our unified framework. Furthermore, we will see how our IMT framework unifies not only the IMT methods, but also other closely related lines of work that are under different motivations and names.

In Chapter 6, we review more related works of IMT by introducing the taxonomy of data-wise training aids (DTA), which are methods that aim to improve model training by optimising training data. As such, we see the role of IMT within this larger body of works on machine learning enhancement.

In the main body of the thesis, we focus on unravelling the narrative of the methods, encompassing their motivation, evolution and connections. While some methods offer nice intuitive interpretations and insightful theoretical results, to maintain a fluid narrative, we reserve these details for Appendix A and B, which we encourage the reader to explore.

## 1.3 Problem Setting

In this work, we consider an IMT problem setting as follows.

In brief, an IMT teacher trains a machine learner who learns using an iterative algorithm by providing samples in each iteration. We treat the teacher and the student as separate entities

who can only communicate through samples. In particular, the teacher cannot "hard-wire" the student's parameters directly.

We consider the student's assets to include:

- A **model** $f$, with a specified architecture, parameterised by $w$

  The model is initialised by $w^0$ before training. During training, the student learns by updating $w^t$ over time.

- A **loss function** $l$

  Given a loss function $l$, the optimal parameter for the student minimises the expected loss

  $$w^* = \arg\min_w \mathbb{E}_{(x,y)\sim\mathbb{P}(x,y)}[l(f(x),y)], \tag{1.1}$$

  where $\mathbb{P}(x,y)$ is the sampling distribution.

- A **learning algorithm**, including a learning rate $\eta_t$ which may change over time

  In this work, we mainly consider the classical gradient descent algorithm:

  $$w^t = w^{t-1} - \eta_t \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \tag{1.2}$$

At each training iteration $t$, the teacher works by providing a sample pair $(x^t, y^t)$ to the student, then the student learns by updating its parameters as (1.2). We refer to $x$ as the input, $y$ as the label.

When providing sample pair $(x^t, y^t)$, the teacher may take some input. The exact input a teacher takes (and is available) will depend on the setting and teaching method, for example, it could be a dataset $\mathcal{D}$, the tracked student's parameter $w^t$, or the optimal parameter $w^*$.

In summary, the iterative teaching-learning procedure is shown in Algorithm 1. To be specific, the teaching methods we will discuss in this work take the place of line 2 in Algorithm 1.

If we compare the IMT setting to normal model training (without a teacher), we can view the IMT teacher as a data feeder with some strategy, so instead of getting *i.i.d.* samples from the sampling distribution, data are now provided in a designed way to guide learning. In fact, under this paradigm, a standard SGD can be seen as a random teacher – the teacher takes a dataset as its input and provides a randomly selected sample in each iteration.[2]

[2] In practice, we often loop through a shuffled dataset to ensure all sample get selected.

**Algorithm 1** Problem Setting

**Input:** student's initial parameter $w^0$, loss function $l$, learning rate $\eta_t$ (over time), number of iterations $T$

**Output:** student's final parameter $w^T$

1: **for** $t = 1, \ldots, T$ **do**
2: $\quad (x^t, y^t) \leftarrow \text{Teacher}(\text{TeacherInput})$ ▷ *Teacher works*
3: $\quad w^t = w^{t-1} - \eta_t \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}}$ ▷ *Student learns*
4: **return** $w^T$

Having the problem now clearly outlined, we are prepared to delve into the exploration of different teaching methods

**Generalisability Note** For simplicity, we will mainly discuss the teaching methods in the case of batch size 1. In general, at each iteration, the student can receive a mini-batch of samples $\{x_j^t, y_j^t\}_{j=1}^m$, where $m$ is the batch size, and update its parameter by

$$w^t = w^{t-1} - \eta_t \cdot \frac{1}{m} \sum_{j=1}^{m} \frac{\partial l\left(x_j^t, y_j^t | w^{t-1}\right)}{\partial w^{t-1}}.$$

For random samples, this is sometimes differentiated from SGD and referred to as mini-batch gradient descent. It is straightforward to extend our method to the mini-batch case.

# Chapter 2

# The "OG" Case

We begin our discussion of various IMT methods with the basic, "OG" case of IMT – omniscient, greedy teaching. It is the first teaching setup considered in the initial work of the IMT series (Liu et al., 2017), and by far the most well-developed set of methods in IMT.

**Chapter Outline** We introduce and motivate the setting of omniscient teaching in Section 2.1, then introduce the greedy teaching strategy in Section 2.2. Specifically, under the greedy strategy, we discuss three teaching methods: the baseline IMT (Liu et al., 2017), LAST (Liu et al., 2021) and DHT (Qiu et al., 2022). By comparing these methods, in Section 2.3, we identify the first dimension of our unified IMT framework – teaching capability. Accordingly, in Section 2.4, we perform experiments to empirically compare different methods along this dimension.

## 2.1 Omniscient Teacher

By omniscient teaching, we assume the teacher has access to the student's full asset. In particular, the teacher can trace the student's parameter $w^t$ over time. The teacher also knows the optimal $w^*$. The objective of an omniscient teacher is to lead the student as closely as possible to the optimal by providing a series of examples:

$$\min_{(x^1,y^1),\ldots,(x^T,y^T)} d\left(w^T, w^*\right), \tag{2.1}$$

where $d(\cdot,\cdot)$ is a discrepancy measure. In this work, we typically consider the squared Euclidean distance $\|\|^2$.

### 2.1.1 Why Care?

At this point, it is necessary to re-emphasise the important question: if the optimal $w^*$ is already known, what is the point of studying IMT?

By studying omniscient teaching, our focus is not on the machine learning problem of learning the optimal model, but on the teacher's problem: how do we maximally influence the student via a series of data? Besides providing intriguing insights into the importance of data in training processes, IMT also sees its application in various practical contexts. For example:

- In education, IMT may help the design of more intelligent online personalised education systems (Xu et al., 2021).
- Though we will mainly consider the teacher's target to be the optimal, loss minimising $w^*$, in general, it does not have to be.

  In cyber-security, a data-poisoning attack is where a hacker attempts to manipulate the behaviour of a machine learning system by maliciously modifying training data. Here, the hacker plays the role of the teacher, and the victim is the unsuspecting student. It will be helpful to understand the mechanisms of these attacks to defend against them (Mei and Zhu, 2015; Alfeld et al., 2016, 2017).

In Chapter 4, we will also discuss other settings of IMT other than omniscient teaching, namely, black-box teaching.

## 2.2 Greedy Teacher

In general, it is hard to optimise (2.1) exactly. As a common approximate strategy, a greedy teacher minimises the difference between the student's parameter and the optimal parameter at each iteration

$$\min_{x^t, y^t} d\left(w^t, w^*\right). \tag{2.2}$$

More specifically, using the squared Euclidean distance, the parameter discrepancy decomposes as

$$\begin{aligned}\left\|w^t - w^*\right\|^2 &= \left\| w^{t-1} - \eta_t \cdot \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} - w^* \right\|^2 \\ &= \left\|w^{t-1} - w^*\right\|^2 + \eta_t^2 \cdot \left\| \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \right\|^2 - 2\eta_t \cdot \left\langle w^{t-1} - w^*, \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \right\rangle .\end{aligned}$$

That says, at iteration $t$, the teacher provides sample pair $(x^t, y^t)$ that maximally minimises $\|w^t - w^*\|^2$ compared to $\|w^{t-1} - w^*\|^2$. (2.2) is equivalent to

$$\min_{(x^t, y^t)} \eta_t^2 \cdot \underbrace{\left\| \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \right\|^2}_{T_1\left(x^t, y^t, | w^{t-1}\right)} - 2\eta_t \cdot \underbrace{\left\langle w^{t-1} - w^*, \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \right\rangle}_{T_2\left(x^t, y^t, | w^{t-1}\right)} . \tag{2.3}$$

For notational convenience, we define

$$G\left(x^t, y^t | w^{t-1}\right) = \eta_t^2 \cdot T_1\left(x^t, y^t | w^{t-1}\right) - 2\eta_t \cdot T_2\left(x^t, y^t | w^{t-1}\right). \tag{2.4}$$

We read $G\left(x^t, y^t | w^{t-1}\right)$ as the score of sample $(x^t, y^t)$ for student $w^{t-1}$ at iteration $t$, which we wants to minimise.[1] In fact, $T_1$ and $T_2$ also have nice intuitive interpretations as example difficulty and usefulness (Liu et al., 2017). We leave the discussion of this intuition to Appendix A.

### 2.2.1 Sample Selection

With the goal of optimising (2.3), Liu et al. (2017) mainly[2] considered a sample selection method which best matches a normal machine learning scenario: at each iteration, the teacher selects the best sample with the lowest score from a labelled dataset, i.e. $(x^t, y^t) \in \mathcal{D}$. Compared to an SGD teacher who returns a sample randomly, the IMT teacher now selects a sample by criteria. We show this straightforward algorithm in Algorithm 2.

[1] The rather counter-intuitive usage of "minimising score" is to keep consistent with conventions in previous works.

[2] The theoretical study of Liu et al. (2017) also introduced three types of joint synthesis teachers. They have mainly been considered for the purpose of theoretical analysis, and are not the focus of the main method proposed. We include a discussion of the theoretical results in Appendix B.

**Algorithm 2** The baseline IMT at iteration $t$

**Input:** dataset $\mathcal{D}$, student's parameter $w^{t-1}$, loss function $l$, learning rate $\eta_t$
**Output:** sample pair $(x^t, y^t)$

1: **for** $(x, y) \in \mathcal{D}$ **do**
2: Compute $G(x, y|w^{t-1})$ by Equation 2.4
3: $(x^t, y^t) \leftarrow \min_{(x,y)\in\mathcal{D}} G(x, y|w^{t-1})$
4: **return** $(x^t, y^t)$

As the first practical method and later a common baseline in the study of IMT, this omniscient, greedy, sample selection method is often referred to as simply "IMT" in later IMT works (with a slight abuse of terms). In this work, we refer to it as "the baseline IMT method".

#### Limitation

A major drawback of the baseline IMT method is the cost of traversing the whole dataset to find the best example, which scales linearly with the size of the dataset.

To overcome this problem, **Local Sensitivity Teaching (LST)** (Xu et al., 2021) replaces the linear search with a hash table lookup, which takes a constant cost after data pre-processing. More specifically, for certain models and loss functions, LST is able to approximately reformulate (2.3) as an adaptive inner product sampling problem, and then select samples using the efficient local sensitivity sampling (hence the name). However, LST depends on the form of the student model and loss function to be reasonably linearisable.

As another way round, a different direction of work avoids the cost problem by avoiding sample selection altogether – proposed by Liu et al. (2021), **label synthesis teaching (LAST)** falls back to taking random samples from the dataset, but focuses on optimising labels to optimise teaching. We introduce this method in the next section.

### 2.2.2 Label Synthesis Teaching (LAST)

At iteration $t$, a LAST teacher takes a random sample $x^t$ from the dataset and synthesises label $\tilde{y}^t$ by solving the optimisation problem[3]

$$\min_{\tilde{y}^t} \eta_t^2 \cdot \left\| \frac{\partial l\left(x^t, \tilde{y}^t | w^{t-1}\right)}{\partial w^{t-1}} \right\|^2 - 2\eta_t \cdot \left\langle w^{t-1} - w^*, \frac{\partial l\left(x^t, \tilde{y}^t | w^{t-1}\right)}{\partial w^{t-1}} \right\rangle. \tag{2.5}$$

[3]From here on, we use tilde notations for synthesized data and notations without tilde for original data.

We can optionally add constraints to the optimisation problem (2.5) to limit the valid value of $\tilde{y}^t$. For example, some constraints we may consider include:

- One-hot constraint, i.e. $\sum_i \tilde{y}_i^t = 1, \tilde{y}_i \in \{0,1\}$ for all $i$. $\tilde{y}_i^t$ is the $i$-th component of $\tilde{y}^t$.
- Soft constraint, i.e. $\sum_i \tilde{y}_i^t = 1, \tilde{y}_i \geq 0$ for all $i$.
- Magnitude constraint,
  - Around the ground truth label, i.e. $\|\tilde{y}^t - y^t\| < r$, where $r > 0$,
  - Or around the learner's current prediction $\|\tilde{y}^t - f(x^t)\| < r$.

In algorithmic form, the LAST method is shown in Algorithm 3.

**Algorithm 3** Omniscient, Greedy LAST at iteration $t$

**Input:** dataset $\mathcal{D}$, student's parameter $w^{t-1}$, loss function $l$, learning rate $\eta_t$
**Output:** sample pair with synthesised label $(x^t, \tilde{y}^t)$

1: Randomly select $x^t$ from $\mathcal{D}$
2: Synthesise $\tilde{y}^t$ by solving Equation 2.5
3: **return** $(x^t, \tilde{y}^t)$

#### Motivation from soft label methods

Besides avoiding the cost of searching through datasets, another motivation behind LAST comes from the success of using soft labels in training: knowledge distillation uses soft labels predicted by a trained model (Hinton et al., 2015), self-training uses soft labels predicted by the current student (Xie et al., 2020), label smoothing (Szegedy et al., 2016) and DisturbLabel (Xie et al., 2016) both modify labels by some heuristic. In one way or another, they have shown that using the ground truth hard label in training is not necessarily optimal. We refer to Chapter 6 for more discussion of related works of IMT.

#### Interpretation as sample re-scaler

When not using one-hot or soft constraints, it is possible that the optimal solution to (2.5) is out of a semantically sensible range for a label value. How do we interpret the synthesised "label" in these cases?

Liu et al. (2021) proposed a view that we can see LAST as a sample re-scaler. It is easy to see the case when $y$ is a scalar: Modifying the label is essentially to re-scale the gradient without

changing the direction. Moreover, in the case of mini-batch training, modifying labels is more flexible than adjusting the learning rate as it is sample-specific. When $y$ is a vector, we can generalise this idea and think of LAST as a multi-dimensional sample re-scaler. Overall, allowing less constrained $\tilde{y}^t$ entitles the teacher to greater influence over the student in one iteration.

### 2.2.3 Data Hallucination Teaching (DHT)

A LAST teacher synthesises labels for randomly selected inputs. Following LAST, Qiu et al. (2022) proposed data hallucination teaching (DHT) which does the opposite – for a randomly sampled label $y^t$, a DHT teacher synthesises an input $\tilde{x}^t$ by solving

$$\min_{\tilde{x}^t} \ \eta_t^2 \cdot \left\| \frac{\partial l\left(\tilde{x}^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \right\|^2 - 2\eta_t \cdot \left\langle w^{t-1} - w^*, \frac{\partial l\left(\tilde{x}^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \right\rangle. \tag{2.6}$$

Like before, we may add constraints on the value of $\tilde{x}^t$. For example, for image input, a natural constraint to consider is the pixel space $\tilde{x}^t \in [0,255]^d$, where $d$ is the dimension of input data.

One may expect a DHT teacher to be more powerful than a LAST teacher with its control over the input space. However, the Qiu et al. (2022) did not give a direct comparison in their experiments. We will answer this pending question in Section 2.4, but before that, we would like to take a moment to reflect: by comparing the three methods we have introduced, we are ready to identify the first dimension of our unified IMT framework – teaching capability.

## 2.3 Dimension 1: Teaching Capability

By teaching capability, we refer to the range of examples a teacher is capable of providing. It is directly related to the teacher's teaching space, i.e. the space from which the teacher can provide examples.

We inherit the use of the term from Liu et al. (2017, 2021), where it is used to refer to the constraints on the teacher's optimisation problem. Here, by comparing the different methods, we allow it to have a more general meaning. Specifically, there are two levels of information it carries:

1. Which part of the data does the teacher have control over (rather than taking random samples)? For example,
    - A LAST teacher can control the label for a given input $(y|x)$
    - A DHT teacher can control the input for a given label $(x|y)$
    - A baseline IMT teacher has control over the whole sample pair (jointly $x, y$)
2. To what degree of freedom can it control the data? For example, can it:
    - Make a choice from a dataset? E.g. the baseline IMT.
    - Provide whatever they want, even regardless of semantic meaning? E.g. unconstrained LAST or DHT.
    - Or somewhere in the middle? E.g. constrained LAST or DHT.

      To provide a clearer understanding, other examples of this category we will see later in this thesis include:
        - A combination-based teacher (Section B.2.1) can synthesise inputs that is a linear combination of inputs from a candidate pool.
        - A generative DHT (Section 3.1) synthesises input following the underlying distribution of the original dataset to some degree.
        - A *mixup*-based teacher (Section 3.1) can provide a sample pair by interpolating between two randomly selected samples from the dataset.

The organisation of some selected methods discussed in this work in the teaching capability space is visually presented in Figure 2.1.

Some hybrid methods that have been explored in previous works also fit nicely into this framework: for example, the "IMT + LAST" method (Liu et al., 2021), which first selects a sample by the baseline IMT then optimises the label using LAST, can be thought of as optimising $y$ based on the optimal of a discrete set of points. "DHT + LAST" (Qiu et al., 2022) first optimises $x$ given $y$ then optimises $y$ given $x$, which may approximate a joint optimisation. Theoretically, an unconstrained joint synthesis teacher should have the greatest teaching capability of all.

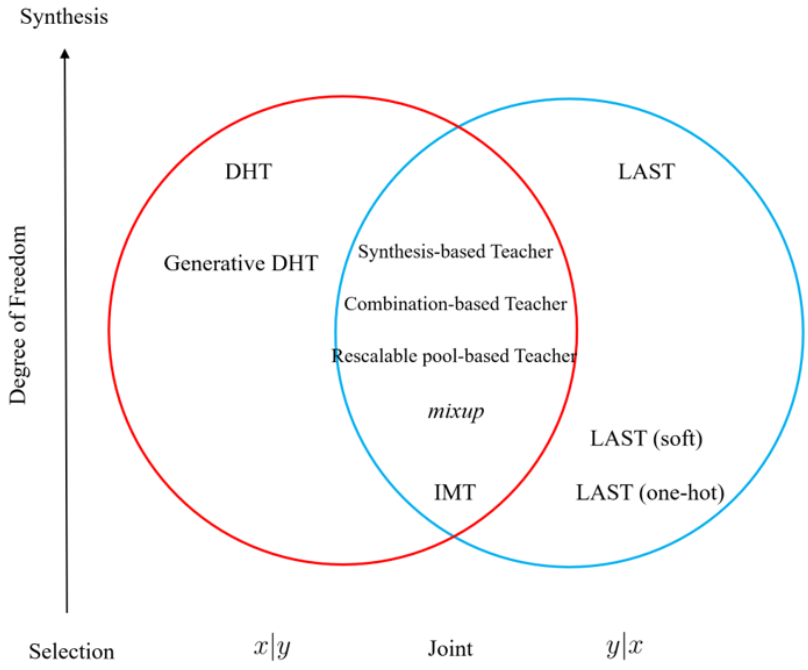



Fig. 2.1 Organisation of selected teaching methods in the teaching capability space. IMT stands for the baseline IMT. When unspecified, LAST and DHT are unconstrained. Generative DHT and *mixup*-based teacher are discussed in Section 3.1,

## 2.4 Experiments: Comparing across Teaching Capability

### 2.4.1 Experimental Setup

For our experiments, we consider a linear logistic regression learner trained for a binary classification task. We consider two datasets – a more intuitive, synthetic half-moon dataset, and a projected MNIST dataset as a harder dataset. We train each model for 1000 iterations. All experiments are repeated 10 times with different seeds. For full experimental details, we refer to Appendix C.

**Teaching methods** Excluding SGD as a baseline, we experiment with 4 teaching methods: the baseline IMT (joint select), LAST (synthesise $y|x$), DHT (synethesis $x|y$) and a joint synthesis teacher. For each of the synthesis methods, we experiment with an unconstrained version and a constrained version. For the constrained method, we constrain the synthesised data into a semantically sensible range: We constrained the synthesised $y$ to be a soft label, and $x$ to be in the range of the original data (“xrange”).

**Performance metric** For each teaching method, we use three metrics to evaluate the convergence performance: the student's parameter discrepancy measured in squared Euclidean distance $\|w^t - w^*\|^2$, loss on the training set and classification accuracy on the test set.

#### Optimisation Method

We make a general note on the optimisation methods we could use, which depend on the teaching capability constraints and the corresponding type of optimisation method formed:

- For IMT, we search over the dataset.
- For unconstrained optimisation, in general, we could use gradient descent algorithms.
- For optimisation with bounded constraints, e.g. magnitude constraints or box constraints, we could use the projected gradient descent algorithm (Madry et al., 2017).
- For LAST, the optimisation is often convex, so we can use convex optimisation methods to find an exact solution.
- For one-hot constraints, it becomes an integer programming problem. Alternatively, if the number of classes is small, we can simply search over each possibility.

In our experiments, we use gradient descent and PGD for all LAST, PGD and joint synthesis teacher for fairness.

### 2.4.2 Results

The experiment results from the four combinations (two datasets, unconstrained vs constrained) are shown in Figure 2.2-2.5.

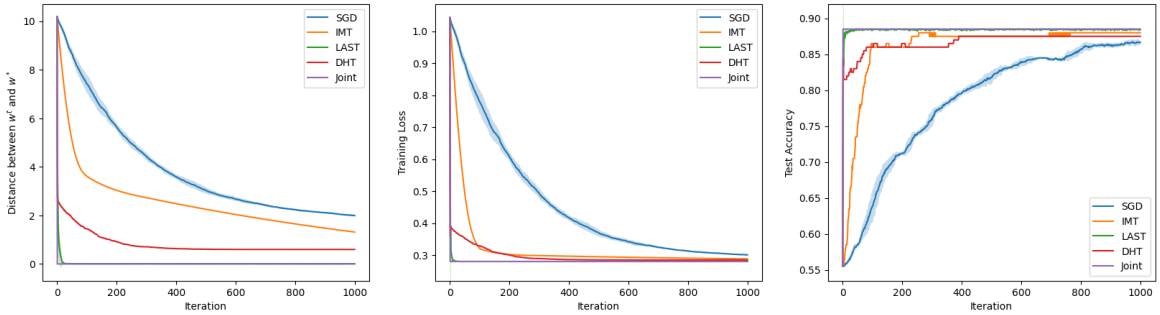



Fig. 2.2 Unconstrained Teaching on Half-moon Dataset

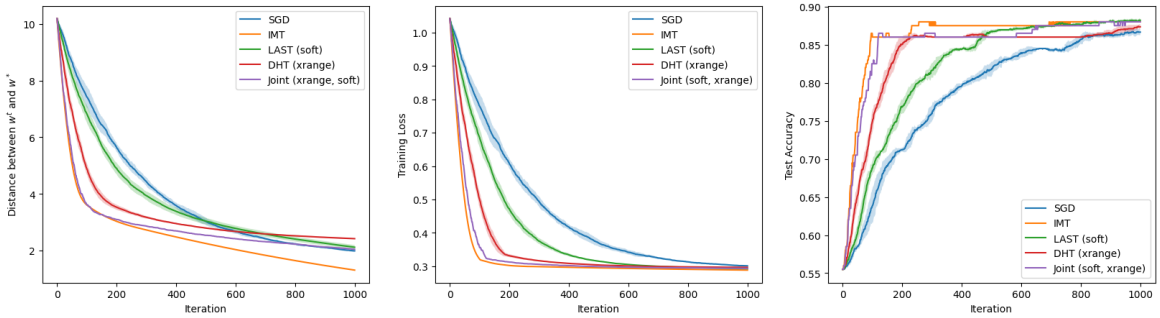



Fig. 2.3 Constrained Teaching on Half-moon Dataset

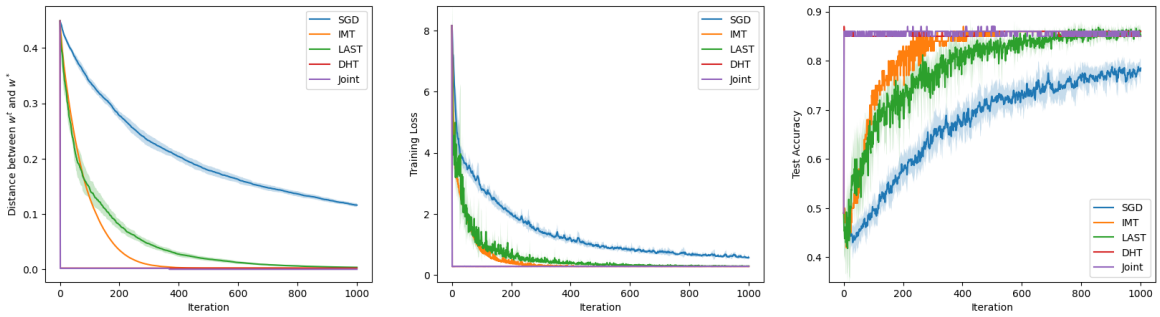



Fig. 2.4 Unconstrained Teaching on MNIST Dataset

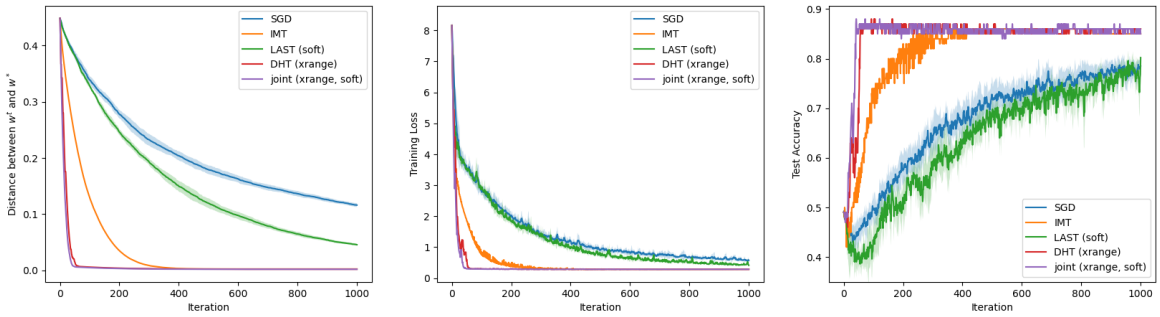



Fig. 2.5 Constrained Teaching on MNIST Dataset

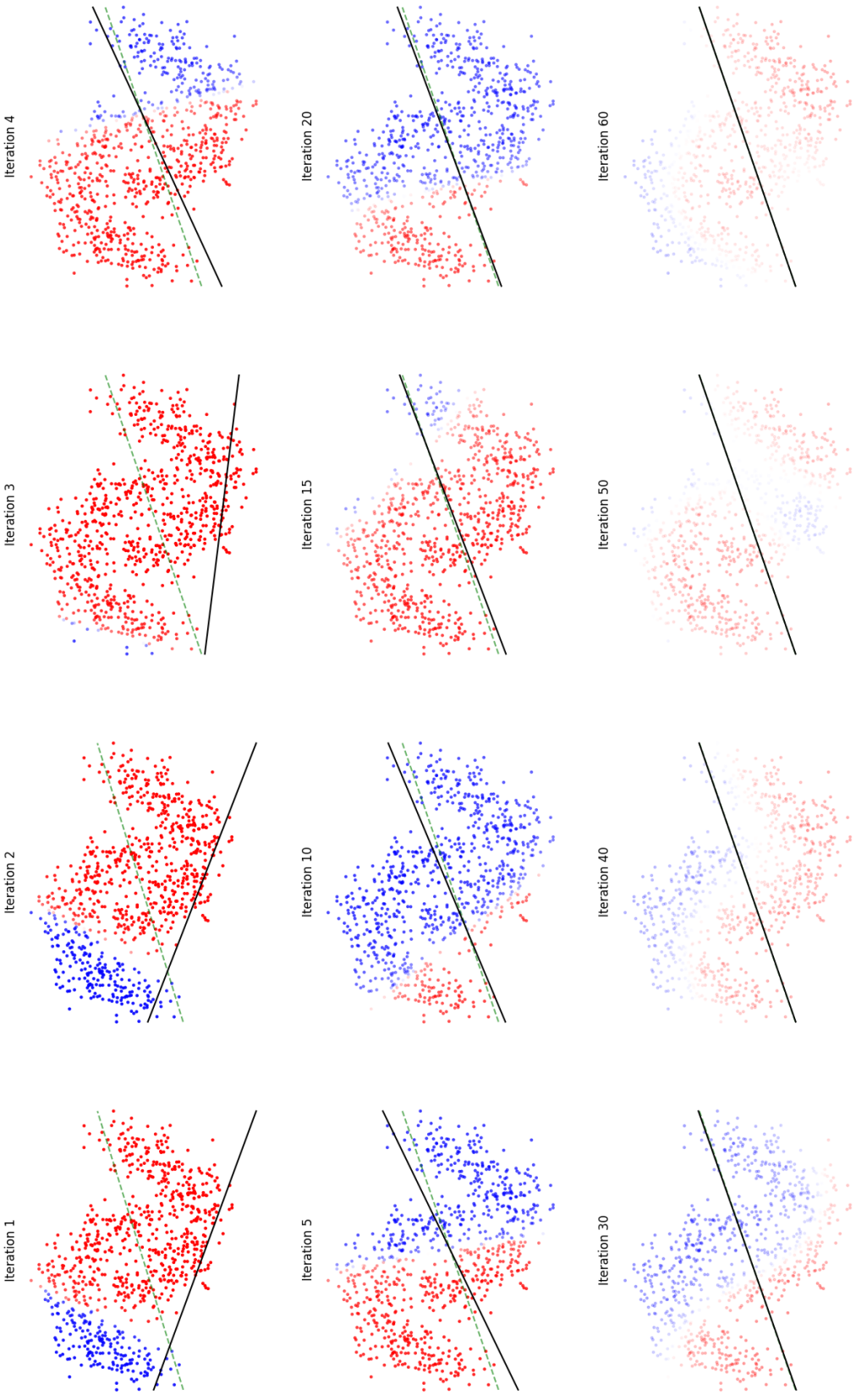



Fig. 2.6 Label synthesised on half-moon data by LAST teacher at different iterations over the training process. Red stands for positive value, blue stands for negative value. Transparency stands for magnitudes. The green line shows the optimal classifier. The black line indicates the current status of the student.

### LAST vs. DHT vs. Joint

In general, we are able to observe that Joint > DHT > LAST as we expect, except for the unconstrained LAST on the half-moon dataset, which showed almost equivalent performance as a joint synthesis teacher.

Investigating into this case, we visualise the label synthesised by LAST on the dataset in Figure 2.6. We can see that on this simple dataset, there exists a simple label synthesising strategy for the LAST teacher, which depends on the interception of the current student and the optimal model. Thus, the LAST teacher has been able to exploit this solution since the start of training, leading to an aggressive convergence.

On the other hand, the optimisation problem for the DHT teacher may be harder to solve, as $x$ has a more complex involvement in the loss gradient through the model, resulting in that it cannot outperform LAST in this case. In other cases, we see a DHT teacher does benefit from the more flexible teaching space of $x$. On the MNIST dataset, it achieves performances near a joint synthesis teacher.

### Unconstrained vs Constrained

For all synthesis teachers, the performance degrades as we move from unconstrained to constrained. The phenomenon is more obvious on the half-moon dataset. We argue that this is because the optimising solution is easier to find on this simple dataset, allowing the synthesis teachers to exploit this benefit when they are unconstrained. Also, when constrained, the space of the half-moon dataset is relatively small. On the other hand, finding the optimal solution on the harder and higher-dimensional MNIST dataset is always harder.

### IMT vs. Synthesis

On the half-moon dataset, the performance of all synthesis teachers is greatly decided by the magnitude constraints.

On the MNIST dataset, the performance of IMT lies between LAST and DHT. We can interpret this as the flexibility gained by synthesising in the relatively narrower label space is not enough to compensate for the shortcomings of a random, non-optimal sample.

For most real-world datasets, we should expect the observation on the MNIST dataset to generalise better.

## 2.5 Summary

In this chapter, we have focused on the basic case of IMT: omniscient, greedy teaching and introduced three different methods. By comparing these methods, we identified the first dimension of our unified IMT framework – teaching capacity. Accordingly, we organised a structured set of experiments to compare methods across the teaching capability dimension.

### Limitation

An obvious shortcoming of the greedy methods in this chapter is the potential sub-optimality from the greediness. In fact, this problem has been shown to be substantial in certain cases by Lessard et al. (2019) by reformulating the objective of the omniscient teacher (2.1) as a time-optimal control problem and solving using tools from optimal control theory.

To alleviate this problem, in theory, we could extend the one-step greedy method by considering a longer planning scope $1 < v < T$ and optimising

$$\min_{(x^{t+1},y^{t+1}),\ldots,(x^{t+v},x^{t+v})} \|w^{t+v} - w^*\|^2 \tag{2.7}$$

The problem now is a trade-off between efficiency and optimality... Can we do something better?

# Chapter 3

# Parametrised Teacher

In Chapter 2, we have discussed the greedy strategy for omniscient teaching, as well as its limitation due to this greediness. In seek of an alternative teaching strategy, Liu et al. (2021) proposed a parameterised teacher – a teacher that is implemented as a neural network. Conceptually, we would like to have the teacher network take the relevant inputs and output the optimal example in each iteration of the student's training. Before it is applied to the student, the teacher itself will be trained in a number of "teaching trials".

**Chapter Outline** In this chapter, we discuss and investigate the feasibility of parametrised teaching. We first introduce the formulation and methods for a parametrised teacher in Section 3.1. In Section 3.2, we perform experiments with a basic parameterised DHT teacher (Qiu et al., 2022). Our experiments have shown non-trivial practical nuances of the parameterised method and limitations that have not been noted in previous works.

## 3.1 Methodology

Parameterised teaching has been studied for both LAST (Liu et al., 2017) and DHT (Qiu et al., 2022) and, in theory, can be easily extended to joint synthesis teachers as well. The formulation and techniques used in each method are very similar. Here, we focus on a parameterised LAST teacher for our discussion and make notes on other cases as necessary.

We assume a parametrised LAST teacher synthesises labels using a policy network $\pi$ parameterised by $\theta$,

$$\tilde{y} = \pi_\theta(x, y, w, w^*).$$

Similarly, for a parametrised DHT teacher,

$$\tilde{x} = \pi_\theta(x, y, w, w^*).$$

## Student Training

When applied in the student's training, at iteration $t$, a parameterised LAST teacher takes a random sample $(x^t, y^t)$, the student's parameter $w^{t-1}$ and the optimal $w^*$ to synthesises label $\tilde{y}^t$ using $\pi_\theta$. This procedure is summarised in Algorithm 4.

**Algorithm 4** Parameterised LAST at iteration $t$

**Input:** dataset $\mathcal{D}$, student's parameter $w^{t-1}$

**Output:** sample pair with synthesised label $(x^t, \tilde{y}^t)$

1: Randomly select $(x^t, y^t)$ from $\mathcal{D}$

2: $\tilde{y}^t \leftarrow \pi_\theta(x^t, y^t, w^{t-1}, w^*)$

3: **return** $(x^t, \tilde{y}^t)$

We note that in Algorithm 4, the parameterised LAST teacher no longer takes the student's loss and algorithm in its input (compared to the greedy LAST in Algorithm 3), as we assume the teacher has learned the required information during the teacher's training.

## Teacher Training

During the teacher's training, the training objective is

$$\begin{aligned} &\min_\theta \left\| w^T(\theta) - w^* \right\|^2 \\ &\text{s.t. } w^T(\theta) = \arg\min_w \mathbb{E}_{(x,y)} \left[ l\left(x, \pi_\theta(x, y, w, w^*) | w\right) \right], \end{aligned} \tag{3.1}$$

where the outer problem is objective of an omniscient teacher, the inner problem is the student's problem that is solved by gradient descent.

To solve this bi-level optimisation problem, two types of techniques have been explored:

### 1) Unrolling

We can unroll $v$ steps of gradient descent of the student's problem into the teacher's objective, which is the same technique that has been used in recurrent neural network training (Werbos,

1988), hyperparameter optimisation (Maclaurin et al., 2015) and meta-learning (Finn et al., 2017). In practice, as it is expensive to unroll a large number of steps, we may choose $1 < v < T$. When $v = 1$, we recover the case of greedy teacher.

Explicitly, the problem now becomes

$$\begin{aligned} &\min_{\theta} \|w^v(\theta) - w^*\|^2 \\ &\text{s.t. } w^v(\theta) = \arg\min_{w} \mathbb{E}_{(x,y)}\{l(x, \pi_\theta(x,y,w,w^*)|w\}. \end{aligned} \tag{3.2}$$

**Practical Nuance** In the context of parameterised DHT, Qiu et al. (2022) proposed that it is helpful to add an auxiliary term to the teacher's objective:[1]

$$\begin{aligned} &\min_{\theta} \|w^v(\theta) - w^*\|^2 + \alpha \sum_{i=1}^{v} l\left(\pi_\theta(x^i, y^i, w^i, w^*), y^i | w^*\right) \\ &\text{s.t. } w^v(\theta) = \arg\min_{w} \mathbb{E}_{(x,y)}\{l(x, \pi_\theta(x,y,w,w^*)|w\}. \end{aligned} \tag{3.3}$$

Here, the auxiliary term can be interpreted as a regularisation on the teacher, which says that when classified using the optimal $w^*$, the synthesised image should not induce a very high loss with the original label.

### 2) Policy Gradient

Alternatively, we can cast the teacher training to a reinforcement learning problem and optimise $\pi_\theta$ using policy gradient (Williams, 1992). More specifically, we can define reward signal $r_t = -\|w^t - w^*\|$ at step $t$ and optimise the expected accumulative reward

$$\max_{\theta} J(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_\theta(a|s)}\left[\sum_{t=1}^{T} \gamma^t r_t(s_t, a_t)\right],$$

where $\gamma \in (0,1]$ is a discounting factor, $(s_t, a_t)$ is the state-action pair at the $t$-th iteration. We synthesise $s_t$ based on $x, y, w^t$ and $w^*$. The action space reflects the teacher's teaching capability.

[1] This formulation is different from that in the original paper (Qiu et al., 2022). Based on results from our experiments and upon confirmation with the original author, the formulation presented here is the formulation that is helpful and used to produce the results in their paper.

The gradient for policy updates is given by

$$\nabla_\theta J(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_\theta(a|s)}\left[\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)\sum_{t=1}^{T}\gamma^t r_t(s_t,a_t)\right].$$

**Practical Nuance** Besides standard techniques that help the convergence of policy gradient methods, the exact computation of the state vector and choice of action space are crucial and non-trivial design choices in practice. For example, features that have been considered for the state vector include:

- The parameter displacement vector $w^t - w^*$
- Inner products between the displacement vector and the loss gradient given each possible action (Liu et al., 2021)

For the action space, a natural choice for a LAST teacher is the uniformly discretised label space, e.g. [0,1], [0.5, 0.5] and [1,0] for a binary classification task. Moreover, we could draw inspiration from other studies, for example,

- Both knowledge distillation (Hinton et al., 2015) and label smoothing (Szegedy et al., 2016) synthesis label as $\lambda y + (1-\lambda)p$, where $y$ is the ground truth hard label, $p$ is the soft label predicted by a trained model in the former and is a uniform distribution in the latter.
- *mixup* (Zhang et al., 2017) synthesises sample pair as $\lambda(x_1,y_1)+(1-\lambda)(x_2,y_2)$, where $(x_1,y_1),(x_2,y_2)$ are randomly selected samples from a dataset.

In both cases, we can consider an action space consisting of possible values of $\lambda$, e.g. $\lambda \in \{0, 0.5, 1\}$.

Despite the various practical nuances, an advantage of the policy gradient method is that it is applicable for objective functions that are not differentiable, which may be the case in black-box teaching that will be discussed in Chapter 4.

## Generative DHT

We note we have not yet put constraints on data synthesised by a parameterised teacher, so they do not necessarily maintain semantic meanings.

An intriguing proposal by Qiu et al. (2022) suggests that we can combine parametrised DHT with generative models. Specifically, they have considered two methods:

**GAN-based teacher** In the training of a GAN-based teacher we optimise [2] [3]

$$\begin{aligned}\min_{\theta}\max_{D}\ & \mathbb{E}_{\tilde{x}\sim p_{\pi}}\log\big(1-D(\pi_{\theta}(x,y,w,w^{*}))\big)+\mathbb{E}_{x\sim p}\log\big(D(x)\big)\\ &+\|w^{v}(\theta)-w^{*}\|^{2}\\ \text{s.t. }& w^{v}(\theta)=\arg\min_{w}\mathbb{E}_{(x,y)}\{l(x,\pi_{\theta}(x,y,w,w^{*})|w\},\end{aligned}$$

where $D$ is the GAN discriminator.

Compared to a standard GAN (Goodfellow et al., 2020), $\pi_{\theta}$ is now trained as the generator, and the teacher's input replaces the random noise.

Compared to a parametrised DHT, we are now jointly training a discriminator to close the gap between synthetic data and real data.

**VAE-based teacher** For VAE-based teacher, we use a pre-trained VAE with decoder $p_{\psi}: z\mapsto(x,y)$. We train the teacher by optimising[4]

$$\begin{aligned}\min_{\theta}\ & \mathrm{KL}\big(\pi_{\theta}(x,y,w,w^{*})\|p(z)\big)\\ &+\|w^{v}(\theta)-w^{*}\|^{2}\\ \text{s.t. }& w^{v}(\theta)=\arg\min_{w}\mathbb{E}_{(x,y)}\{l(x,\pi_{\theta}(x,y,w,w^{*})|w\},\end{aligned}$$

where $p(z)$ is the fixed Gaussian prior for latent variable $z$, KL is the Kullback-Leibler divergence.

Essentially, we are now training the teacher to map input information to the latent distribution, then utilises the VAE decoder to map the latent variable to data.

[2] Omitting the auxilary term (3.3) for clarity.

[3] The formulation presented here (and the following interpretation) is different from that in the original paper (Qiu et al., 2022). In particular, there is no random noise $z$ in the teacher's input. Upon confirmation with the original author, the formulation presented here is the version that gave the presented results in their paper.

[4] Qiu et al. (2022) presented a different formulation using a conditional VAE. Both methods are feasible in principle, and we find that the formulation presented here is easier to train in practice.

**Summary** Both generative DHT methods constrain the data synthesised by the teacher to better match the real data distribution, and in fact, can be seen as constraints on teaching capability in a less stringent form.

Moreover, as the VAE-based teacher uses a pre-trained VAE, while the GAN-based teacher uses a discriminator that is trained jointly with the omniscient objective, we may expect the constraint on the VAE-based teacher to be tighter. This is consistent with the observation by Qiu et al. (2022).

## 3.2 Experiments: Practical Challenges

The idea of a parameterised teacher is elegant. However, can we successfully train such a teacher in practice? One can only know in experiments.

Based on the formulation by Qiu et al. (2022), we attempted to train a parametrised DHT teacher on the half-moon dataset using unrolling. We immediately noticed the inherent challenges in teacher training. By very closely following the author's implementation[5], we were able to replicate their experiments. We refer to Appendix C for the experimental details and show the results in Figure 3.1. In particular, we make a direct and complete comparison with the baseline IMT and greedy DHT.

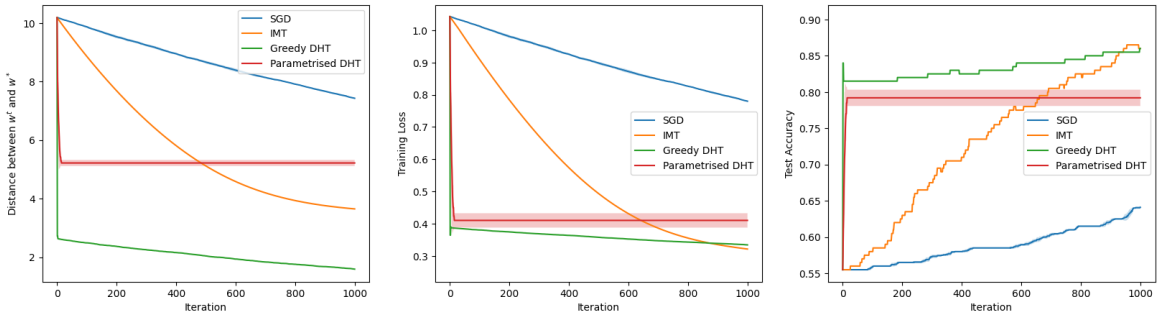



Fig. 3.1 Teaching performance of parametrised DHT teacher on the half-moon dataset

Empirically, we found that though the parameterised DHT can achieve a faster convergence in the first few iterations, outperforming IMT, which is the performance seen in (Qiu et al., 2022), it stops at a local minimum afterwards. IMT, though converged less aggressively at the beginning, actually performs better in the long run when allowed enough number of iterations. On the other hand, the parametrised DHT failed to outperform the greedy DHT either.

[5]Available at https://github.com/Zeju1997/data_halucination_teaching

### Dependency on student initialisation

Moreover, in the implementation, we have noticed other subtlety in the teacher's training, namely, its dependency on the student's initialisation $w^0$.

More specifically, to teach a student with initialisation $w^0$ well in student teaching, it is crucial for the teacher to get trained on students using the same initialisation during teacher training. The teacher's performance will significantly deteriorate if this is not the case, e.g. using random initialisation in training. A comparison of teaching performance is shown in Figure 3.2.

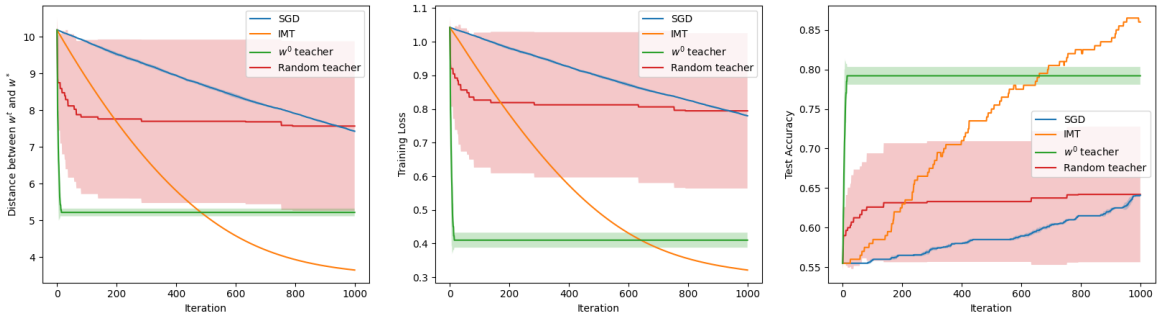



Fig. 3.2 Comparison of teaching performance of a teacher trained for $w^0$ initialisation ($w^0$ teacher) and a teacher trained for random initialisation (random teacher) when training a student initialised at $w^0$. We see the performance of the random teacher is much worse and less stable.

In fact, the issue of heavy dependency on initialisation has been highlighted and discussed in dataset distillation (Wang et al., 2018), a line of work that is very close to IMT and uses similar methods (see Section 5.2). This limitation will greatly restrict the generalisability of a parameterised teacher in practical usage.

## 3.3 Summary

In this chapter, we have discussed parametrised teaching and investigated its feasibility. Overall, our results suggest the difficulties in training and limitations in performance of a parametrised teacher at the current stage. To achieve a fully competent parameterised teacher, significant further effort is still required.

# Chapter 4

# Black-box Teacher

Since Chapter 2, we have been working with an omniscient teacher, who aims to lead the student to the optimal $w^*$, knowing both the student's parameter and $w^*$. While these studies uncover intriguing insights and highlight possible ties to real-world applications, the omniscient assumption remains overly optimistic for numerous practical scenarios. In this chapter, we step out of the omniscient setting and discuss **black-box teaching**.

In fact, the term "black-box teaching" has been considered for different meanings in different literature. Regardless of its interpretation, black-box teaching is widely acknowledged as an extremely challenging task (Liu et al., 2017, 2018, 2021; Qiu et al., 2022). Nonetheless, we will discuss the key ideas and strategies that have been proposed to address this complex issue.

**Chapter Outline** In Section 4.1, we first clarify the two interpretations of black-box teaching that have been considered in literature. For each case, we discuss the proposed teaching strategies in Section 4.2 and Section 4.3 respectively.

## 4.1 What does black-box mean?

Among the different interpretations of "black-box teaching", there is one general agreement – black-box teaching is not the case of omniscient teaching, in which the teacher knows *both* the student's model $w^t$ and the optimal model $w^*$. Therefore, black-box teaching as its negation has been interpreted for two kinds of meanings:

1. The teacher knows the optimal parameter $w^*$, but not the student's parameter $w$.

2. The teacher does not know the optimal parameter $w^*$ either.

While the first case more closely matches the literal meaning of "black box", the second case is more reflective of real-world machine learning situations. In fact, by considering the second case, there is a inclination that IMT's emphasis shifts from conveying a target model to improving the student's generalisability.

Now, we present the strategies that past literature has explored for each scenario.

## 4.2 Case 1: Imitation Teacher

In the first case of black-box teaching, the teacher knows the optimal parameter $w^*$, but cannot access the student parameter $w^t$. This is the case that may be of interest for applications in education or cyber-security (Section 2.1.1). This is also the case that is considered in the initial work of IMT (Liu et al., 2017) and later further investigated in (Liu et al., 2018).

The idea of the proposed solution is simple: The teacher keeps a virtual version of the student which is transparent to the teacher. At the start of each iteration, the teacher first updates the virtual student to mirror the real student by providing test samples to the real student and imitating their behaviour. Then the teacher provides samples based on the information of the virtual student, which is an omniscient teaching problem.

In experiments, the imitation teacher has shown reliable performance. Liu et al. also provided theoretical results on the number of test examples needed. However, they have only considered the case of linear models. Moreover, the method is still based on the assumption that the teacher knows the student's model architecture.

## 4.3 Case 2

In the second case of black-box teaching, where the optimal parameter $w^*$ is unknown to the teacher, two kinds of teaching strategies have been proposed.

### 4.3.1 Surrogate Teacher

A surrogate teacher uses a surrogate objective instead of the parameter discrepancy to train the student.

The term surrogate first occurred in the initial work of IMT (Liu et al., 2017), where the surrogate objective for a greedy teacher is a lower bound of the omniscient objective, assuming a convex loss function.

In later works, a more commonly considered surrogate objective for parametrised teachers is the student's performance on a validation set. For example, for a parametrised LAST teacher with the objective of minimising the student's loss on a validation set:

$$\begin{aligned} &\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_a} l(x,y|w) \\ &\text{s.t. } w(\theta) = \arg\min_{w} \mathbb{E}_{(x,y)\sim\mathcal{D}r}\{l(x,\pi_\theta(x,y,w,w^*)|w\}, \end{aligned} \tag{4.1}$$

where $\mathcal{D}_r$ is the training set and $\mathcal{D}_a$ is the validation set. When the objective is differentiable, we can use the same technique of unrolling or policy gradient introduced in Section 3.1 to train the teaching policy. When the objective is non-differentiable, e.g. accuracy, only policy gradient can be used. They also bear the same practical nuances as discussed, and in general, it is even harder to train a parameterised teacher in black-box settings (Liu et al., 2021; Qiu et al., 2022).

We note that, for the surrogate objectives discussed, while $w^*$ is unavailable, access to the student parameter $w$ is not required either.

### 4.3.2 Performative Teacher

Performative teaching is another method proposed for teaching without $w^*$ but does require $w$. The method is mainly considered for deep neural learners and focuses on teaching the last linear layer. Essentially, at each iteration, a performative teacher inspects the last linear layer $w$ of the student, performs a few steps of gradient descent to learn an optimal $w^*$ and uses it to construct a sample pair. ("The teacher learns in advance.")

The method got its name from its nice connection with performative prediction (Hardt et al., 2022), which is defined as predictions where the action of predicting influences the target they aim to predict. In the case of performative teaching, the teacher looks for an optimal $w^*$ and consequently optimal $(x^t, y^t)$. When the teacher provides the "optimal" $(x^t, y^t)$ to the neural learner, the learner learns and the optimal $w^*$ changes, i.e.

$$\min_{(x^t,y^t)} d(w^t, w^*(t)) \text{ s.t. } w^*(t) \sim \mathcal{M}(x_{t-}, y_{t-1}),$$

where $\mathcal{M}$ is the distribution of the target $w^*$ which depend dynamically on the teaching example.

Interestingly, this rather heuristic method of focusing on the last layer only does lead to better classification results, though by a small margin compared to SGD (Qiu et al., 2022).

## 4.4 Summary

In this chapter, we touched upon the intricate issue of black-box teaching and outlined the strategies that have been proposed.

By now, we have traversed the full series of IMT methods. In the next chapter, we will review the various methods and distill their key dimensions, building towards a unified framework.

# Chapter 5

# A Unified IMT Framework

Having traversed the full series of IMT methods, we are now ready to present our unified IMT framework: By comparing the different methods we have discussed, we identify three key dimensions that underlie the various methods. Thus, we organise the methods as a coherent set of ideas, and each method can be viewed as a combination along the three dimensions and a point in the IMT space.

What's more, our framework has unified not only those works who have named themselves IMT. We found that some other works (Wang et al., 2018; Sucholutsky and Schonlau, 2021; Fan et al., 2018, 2021), though bearing different names and motivations, actually boil down to very similar ideas at the level of mathematical formulation and fit nicely into our framework.

**Chapter Outline** In Section 5.1, we identify the three key dimensions that characterise the IMT space. In Section 5.2, we show how our unified IMT framework also incorporates methods from other lines of work.

## 5.1 Characterising the IMT Space

Recall that, in Section 2.3, we have identified teaching capability as the first dimension of the IMT space by comparing the baseline IMT, LAST and DHT. We now identify the other two dimensions also by comparing and reflecting on different methods.

### 5.1.1 Dimension 2: Teaching Scope

We identify the second dimension by comparing a greedy teacher (Chapter 2) and a parameterised teacher (Chapter 3).

One may think the key difference between a greedy teacher and a parameterised teacher is the number of steps they plan ahead for in teaching – while a greedy teacher optimises the short-sighted one-step goal, a parametrised teacher can be thought of as considering $v$ steps ahead (Section 3.1). However, we discussed in Section 2.5, though it is expensive in practice, it is possible to extend the greedy method to plan for multiple steps.

Then what is the fundamental difference between the two teaching strategies? We think the answer lies in the object that we optimise over: do we directly optimise the sample we provide in student training, or is there a separate teacher training phase where we optimise a sustainable, parametrised teaching policy?[1]

We summarise this idea as the **teaching scope**, which we take as the second dimension of our IMT framework.

### 5.1.2 Dimension 3: Teaching Objective

Finally, we extract the third dimension by contrasting an omniscient teacher and a black-box teacher[2] (Chapter 4): based on the information that is available to them, they optimise different objectives, either the parameter discrepancy if the student's parameter and optimal $w^*$ are available, or a surrogate objective, usually based the student's performance on a validation set.

**Summary** We have now identified the three key dimensions and thus established a unified framework for IMT, which succinctly summarises and organises a variety of IMT methods proposed in a series of works. Now it should be clear to see the connection between different methods, or identify gaps in the existing framework that are worth exploring.

Of course, this organisation will not be exhaustive or definitive, and should be revised as the field progresses.

[1] Interestingly, in this view, a parameterised teacher can be thought of as being far-sighted in another sense.

[2] Here, we mainly consider the surrogate teacher for black-box teaching, as the imitation teacher and performative teacher essentially solve their problem by turning it into an omniscient setting, more specifically, by retrieving $w$ and $w^*$ respectively.

In the rest of this chapter, we will see how two other lines of work that are closely related with IMT fit into this framework.

## 5.2 Unifying beyond IMT

### Dataset Distillation

Dataset distillation (DD)'s line of work (Wang et al., 2018; Sucholutsky and Schonlau, 2021; Bohdal et al., 2020; Such et al., 2020) works around the idea of synthesising a small dataset that gives a comparable training performance as a large dataset.

To see its connection with IMT, we start in our IMT framework, consider optimising directly on a batch of data, say $\tilde{\mathcal{D}} = \{(x_j, y_j)\}_{j=1}^{m}$, assuming it is used repeatedly in a series of $v$ steps, and optimising the loss on a validation set:

$$\begin{aligned} &\min_{\tilde{\mathcal{D}}} \mathbb{E}_{(x,y)\sim\mathcal{D}_a}\left[l(x,y|w^v)\right] \\ &\text{s.t. } w^v = \arg\min_{w} \mathbb{E}_{(x,y)\sim\tilde{\mathcal{D}}}\left[l(x,y|w)\right], \end{aligned} \tag{5.1}$$

we recover the formulation of Wang et al. (2018), the initial work of the DD series. The rest of the series will follow as natural extensions.

It is also interesting to note that the DD series of work has evolved following a very similar line of thoughts on teaching capability as IMT: after the initial work (Wang et al., 2018) which optimises inputs, Sucholutsky and Schonlau (2021) extends DD to jointly optimising input and label, Bohdal et al. (2020) simplifies DD again to optimise label for given input, and Such et al. (2020) proposed to use generative networks.

### Learning to Teach (L2T)

Learning to Teach (L2T) is also a high-level framework, proposed and expanded in a series of works by Fan et al. (2018, 2021). Similar to ours, they also follow the goal of optimising training, but in an even broader sense that they consider all data, loss function (Wu et al., 2018) and student model as optimisable parts.

Despite the ambitious conceptual framework, their practical methods essentially correspond to the parameterised teacher in our IMT framework, trained using policy gradient, optimising loss on a validation set. The teaching space they have considered includes selecting samples

from a dataset (Fan et al., 2018) and reweighting randomly selected samples (Fan et al., 2021).

## 5.3 Summary

We have wrapped up our exploration of the IMT series, presenting it within a cohesive unifying framework and observing its ties to certain related studies. Indeed, IMT is recognized for its extensive affiliations with numerous works. In the subsequent chapter, we will explore these connections in both broader and deeper dimensions.

# Chapter 6

# Related Research Areas

Previous IMT studies have highlighted its close connection with a broad range of research topics: curriculum learning (Bengio et al., 2009), knowledge distillation (Hinton et al., 2015), data augmentation (Krizhevsky et al., 2012; Chen et al., 2020)... just to name a few. From Zhu (2013)'s machine teaching to training techniques emerged in the blooming development of computer vision, how can IMT find its connection with such diverse studies, and how do these studies relate to each other?

We think there is a natural idea that brings everything together: if we take a step back and think generally about the term "machine teaching", it is essentially asking the question of "how can we teach a machine learner better?" More specifically, all these diverse studies fundamentally focus on **facilitating model training with respect to data**. We will refer to them as **data-wise training aids** (DTA).

**Chapter Outline** In this chapter, we collect and review a range of DTA studies, many of which have been noted to relate to IMT before. We first present a taxonomy of the different DTAs in Section 6.1, then give a concise introduction to each of the methods in Section 6.2, highlighting their DTA aspect.

## 6.1 Taxonomy

To classify the different DTA techniques, there are a few key questions we could ask:

**1. At which point does the data handling process take place?**

We divide the methods into three groups:

1. **Static**: Static methods are applied before the iterative training process. Typical static methods include constructing or labelling datasets.
2. **Dynamic**: Dynamic methods are applied during the iterative training process and typically involve modifying some randomly sampled data, e.g. LAST, DHT and *mixup*.
3. **Sequential**: Sequential methods decide the order in which samples are fed to the student, e.g. the baseline IMT, triplet selection.

**2. Which part of the data can we modify? To what degree of freedom?**

This is basically the teaching capability dimension of the IMT framework.

**3. What is the motivation?**

Among the various methods, we have identified broadly two types of goals:

1. **Minimising teaching effort**

   More specifically, this category can be further divided into:

   1.1. Constructing a minimal dataset, which is usually the goal of static methods, e.g. Zhu's machine teaching, dataset distillation.

   1.2. Achieving the minimal number of iterations, i.e. fast convergence. This is usually the goal of dynamic or sequential methods, e.g. IMT.

2. **Generalisability of the trained model**

   Notably, while generalisability traditionally refers to the ability of a model to perform well on unseen data, in recent years, it has been interpreted in a second way as the ability to perform *not unwell* on unseen data, in particular, against adversarial attacks, e.g. (Goodfellow et al., 2014).

   In our taxonomy, we do not differentiate further between the two implications of generalisability. However, we will note that some methods have a particular emphasis on certain types of generalisability.

In Section 6.2, we review a list of DTA studies. The list of topics reviewed is presented in Figure 6.1, primarily ordered by their motivation. Besides, based on the first two dimensions of the taxonomy, a categorisation of these methods, along with methods from the previous chapters, is shown in Table 6.1.

| | **Static** | **Dynamic** | **Sequential** |
|---|---|---|---|
| **Synthetic** | Dataset Distillation<br>Dataset Condensation | LAST<br>DHT | |
| **Modify** | Data Augmentation<br>Knowledge Distillation<br>Label Smoothing<br>Self-training | *mixup* | |
| **Select** | Coreset | | IMT<br>Curriculum Learning<br>Triplet selection |

Table 6.1 Categorisation of DTA methods based on the first two dimensions of the taxonomy. Red highlights methods that only change inputs, and blue highlights methods that only change labels.

## 6.2 Methods Synopsis

### 6.2.1 Training Set Cardinality

The common objective of all methods in this category is to find a smaller dataset that can achieve comparable performance to a larger dataset. Machine teaching (MT) and dataset distillation (DD), which we have introduced in previous chapters, belong to this category.

**Coreset** Coreset studies methods for *selecting* a dataset as the subset of the original dataset.

Like MT, the study of coreset started theoretically (Agarwal et al., 2005), considering specific models such as SVM (Tsang et al., 2005) and k-means (Har-Peled and Kushal, 2005), and later moved to practical algorithms (Huggins et al., 2016; Campbell and Broderick, 2018; Mirzasoleiman et al., 2020).

**Dataset Condensation (DC)** Like DD, DC (Zhao et al., 2020; Zhao and Bilen, 2021) constructs a dataset by synthesis. While DD synthesises data to optimise the training target

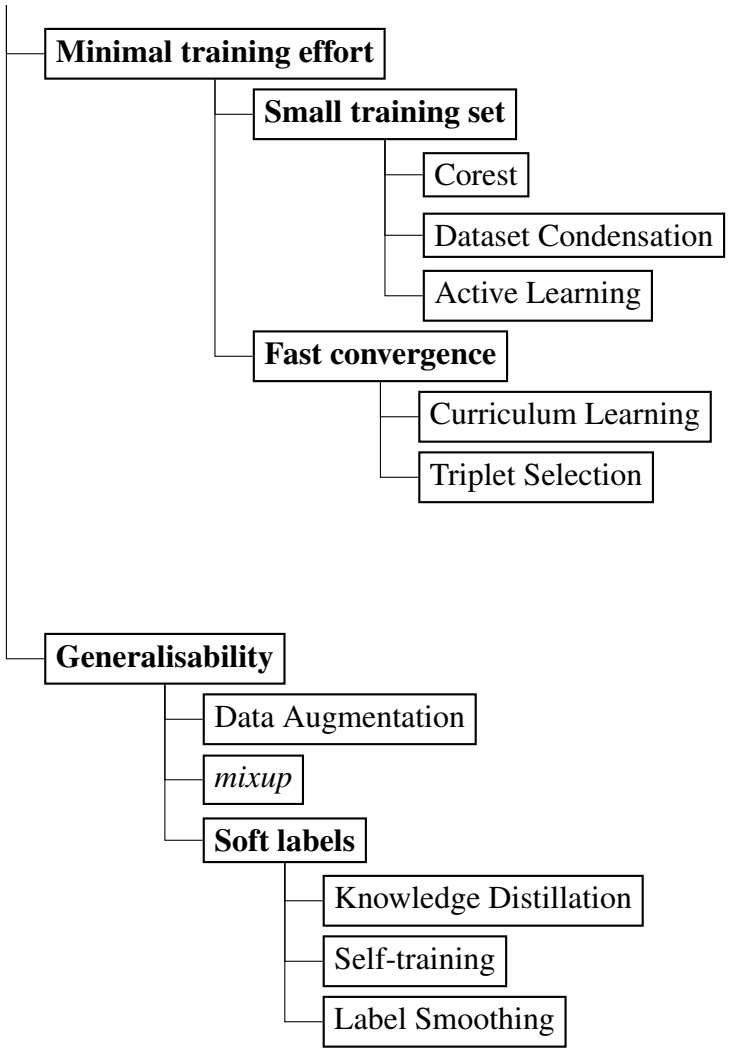



Fig. 6.1 List of DTA studies reviewed

directly (Section 5.2), DC mimicks specific behaviour of the original dataset, e.g. the gradient induced on a model.

**Active Learning** Active learning (Settles, 2009; Sener and Savarese, 2017), as a high-level concept, lets the student actively query the teacher for labels at new inputs, exploring the input space at their pace. There have been different specific query strategies, Settles (2009) provides a comprehensive survey.

Unlike other static methods aiming for training set cardinally, active learning takes place as the student learns. However, we still categorise it here because the most common setting that active learning is considered is during the dataset construction process, where it helps reduce the teacher's effort for labelling inputs. We did not include active learning in Table 6.1 as it is hard to categorise.

When compared with IMT, active learning can be seen as choosing samples in training using only the student's information. In comparison, an IMT teacher knows both the training goal and the student's status to guide the student accordingly. So in general, we can expect a smaller sample complexity for an IMT teacher. A similar result for MT has also been shown by Zhu (2013).

### 6.2.2 Convergence Speed

**Curriculum Learning** Curriculum learning is a general training strategy that suggests examples should start simple and gradually increase in difficulty. Bengio et al. (2009) brought this idea to machine learning from cognitive science and showed some toy experiment demonstrations constructed according to heuristics. This idea has also been seen effective in other studies, both empirically (Schroff et al., 2015) and intuitively (Liu et al., 2017, 2021).

**Triplet selection in FaceNet training** Schroff et al. (2015) proposed triplet selection as an add-on to FaceNet, a contrastive learning system using the triplet loss, which takes a triplet of samples for each loss computation.

The idea of triplet selection is to choose hard examples – examples that incur greater loss. However, the authors made the empirical note that using the hardest examples can lead to local minima in early stages of training or even a collapsed model. In practice, they have used the so-called semi-hard examples. This aligns with the intuition curriculum learning (Bengio et al., 2009).

### 6.2.3 Generalisability

**Data Augmentation** In general, data augmentation methods train the student model with similar but different examples for generalisability. More specifically, it is commonly used to refer to techniques in computer vision such as reflections, rotations and color distortion (Chen et al., 2020).

We note that the use of data augmentation in practice can be either static or dynamic. In Table 6.1, we categorise it as a static method as it can be performed beforehand training.

***mixup*** In a nutshell, *mixup* (Zhang et al., 2017) proposes to train the model with virtual examples constructed as

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j$$

where $(x_i, y_i), (x_j, y_j)$ are two random samples from the dataset, $\lambda \sim \text{Beta}(\alpha, \alpha)$ where $\alpha \in (0, \infty)$.

Intuitively, *mixup* encourages linear behaviour between available training examples, thus improving the generalisability of the trained model, in particular, against adversarial attacks.

#### Soft Label

**Knowledge Distillation** Knowledge distillation (Hinton et al., 2015) studied methods for "distilling" the knowledge of a large, cumbersome model to a lightweight model. More generally, knowledge distillation serves as a technique for model compression (Buciluǎ et al., 2006), whose goal is to reduce model sizes without loss of performance.

Specifically, Hinton et al. proposes to train the student model using soft label information from the teacher:

$$\tilde{y}_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)},$$

where $\tilde{y}_i$ represent the probability for class $i$, $z_i$ is the $i$-th logits before the final softmax layer of the teacher, and $T$ is temperature, Normally, $T$ is set to 1. In training, we use a higher temperature to produce a softer probability distribution (thus the name).

**Label smoothing** The label smoothing technique used in (Szegedy et al., 2016) avoids the use of hard labels, which are argued to cause overfitting. It "smooths out" the label

heuristically by

$$\tilde{y} = \mu y + \frac{1-\mu}{K}\mathbf{1},$$

where $K$ is the number of classes.

**Self-training** In self-training (Xie et al., 2020), the student is trained on a mixture of teacher-labelled data and data with pseudo-labels predicted by the student from the last iteration.

## 6.3 Summary

In this chapter, we have viewed IMT as a method in an even larger taxonomy of data-wise training aids. As a result, we are able to gain a more holistic comprehension of IMT's role and its ties to closely related areas of work.

# Chapter 7

# Conclusion

In this thesis, we reviewed the latest developments in the newly emerged topic of iterative machine teaching (IMT), which forms part of the large body of works on comprehensive studies of machine learning. We examined the key line of IMT methods, identified three key dimensions that underlie the diverse method formulations and provided a unified framework. Guided by the first dimension of teaching capability, we performed a set of benchmarking experiments for different methods of omniscient, greedy teaching. On the second dimension, we introduced parametrised teachers, but our experiments showed their current limitations in practice. On the third dimension, the problem of black-box teaching is even more challenging, and we outlined the various strategies proposed. Beyond discussing methods within the IMT paradigm, we also viewed IMT in the larger taxonomy of studies that assist model training data-wise. Overall, we hope these two levels of review and organisation provide a succinct summary and clear view of IMT and related methods, making explicit links and highlighting problems and gaps for future researchers.

## 7.1 Open Questions

Finally, we want to end this thesis with some further problem dimensions that have not been studied in existing IMT works, but will be interesting to consider in the future.

**Learning Algorithm** We have been mainly working with a student who learns using a basic gradient descent algorithm (Equation 1.2). There are other optimisers that are commonly used nowadays, especially momentum-based algorithms such as Adam (Kingma and Ba,

2014). The current teaching methods have not been optimised for them. We may expect it to be especially non-optimal for greedy teachers. Though we can naively apply a parameterised teacher, we have already seen the difficulty of training and its limited performance in practice.

**Strategic Student** The IMT problem setting we worked with considered a passive student: the student takes and updates its parameter with every sample provided. However, this may not best reflect the case of some real-world scenarios, for example,

- In education, a human student may aim to pass a test while minimising their effort, thus skipping exercises that seem too easy or too challenging.
- In cyber-security, an intelligent system may recognise potential malicious data and proactively implement countermeasures to neutralize the threat (Mei and Zhu, 2015; Alfeld et al., 2016, 2017).

In these cases, the students are strategic individuals who have their own set of actions and goals, which may not necessarily align with the teacher's objective. We conjecture that such settings can be framed as a game between the teacher and the student, giving rise to the study of **game-theoretical machine teaching**.

There have been some machine teaching works that consider active students (Peltola et al., 2019; Yuan et al., 2021). However, they have considered cooperative students, who share the same goal as the teacher. Also, Yang et al. (2018); Wang et al. (2019, 2020a,b) provide a statistical framework for cooperative communication.

On the other hand, notions from game theory have been utilised in machine learning in strategic classification (Hardt et al., 2016) and performative prediction (Perdomo et al., 2020). However, their settings are different from ours in that both parties (the classifier and the classified subject) are "mature" individuals with set behaviour. There is no notion of "teaching and learning" taking place.

# Appendix A

# Intuitive Interpretation

In this section, we discuss a nice intuitive interpretation of the teaching strategy of an omniscient, greedy teacher: In training, we should follow a "easy-to-hard" principle (Liu et al., 2017, 2021). This assertion is exactly the core idea of curriculum learning (Bengio et al., 2009) and has also been seen in other settings (Kumar et al., 2010; Schroff et al., 2015).

To recap, at each iteration, an omniscient, greedy teacher optimises

$$\min_{(x^t,y^t)} \eta_t^2 \cdot \underbrace{\left\| \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \right\|^2}_{T_1\left(x^t,y^t,|w^{t-1}\right)} - 2\eta_t \cdot \underbrace{\left\langle w^{t-1} - w^*, \frac{\partial l\left(x^t, y^t | w^{t-1}\right)}{\partial w^{t-1}} \right\rangle}_{T_2\left(x^t,y^t,|w^{t-1}\right)} \tag{A.1}$$

We have defined $T_1$ and $T_2$, which we will now interpret respectively as the **difficulty** and **usefulness** of example $(x^t, y^t)$ for student $w^{t-1}$. A visual representation that will help the explanation along the way is shown in Figure A.1.

**Difficulty** The difficulty of an example $(x, y)$ is the squared magnitude of the gradient the example pair induced for the model. Intuitively, a harder example would induce a higher loss, a large gradient and consequently an aggressive update.

This interpretation can be most easily seen with some simple examples (Liu et al., 2017): consider data living on the surface of a sphere, i.e. $\|x\| = 1$, then

- For linear regression with squared loss $\frac{1}{2}(\langle w, x\rangle - y)^2$, $T_1 = (\langle w, x\rangle - y)^2$.

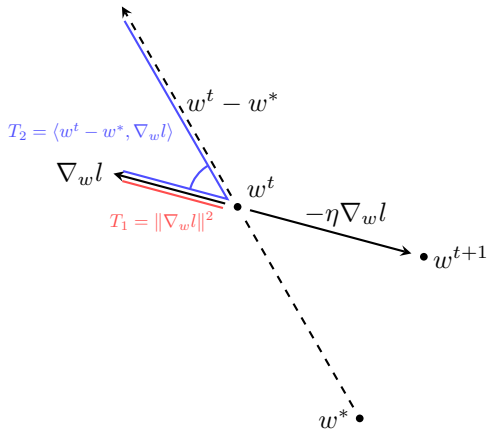



Fig. A.1 A visual representation of example difficulty $T_1$ and usefulness $T_2$ in the parameter space

- For logistic regression with logistic loss $\log(1+\exp(-y\langle w,x\rangle))$, $T_1 = \left\| \frac{1}{1+\exp(y\langle w,x\rangle)} \right\|^2$, i.e. the probability of predicting the wrong label.

**Usefulness** $T_2$ quantifies the usefulness of an example. This can be clearly seen in Figure A.1: $T_2$ as a correlation encaptures the alignment between $\nabla_w l$ and $w^t - w^*$. A higher correlation means the sample induces a gradient in a "righter" direction.

Notably, while our example usefulness is related to the optimal $w^*$, the example difficulty is not. This is different from the heuristic used to construct demonstrative experiments in curriculum learning (Bengio et al., 2009), but intuitively makes sense – we find an example simple or hard based on our own ability, not how true an example is.

Overall, our optimisation goal (A.1) is saying that we will always prefer useful and easy examples. However, we may expect that a useful example is often hard. More explicitly, if we let $\nabla_w l = r \cdot \mathbf{u}$, where $r$ is the magnitude and $\mathbf{u}$ is a unit vector indicating the direction, then the optimising goal becomes

$$\min_{x,y} \eta^2 \underbrace{r^2}_{=T_1} - 2\eta \underbrace{r\langle w^t - w^*, \mathbf{u}\rangle}_{=T_2}. \tag{A.2}$$

If an example has a high $T_2$, it is likely to benefit from a large $r$ and thus also have a high $T_1$. The teacher will always try to achieve a balance between $T_1$ and $T_2$. However, in later phases of training, when the student is behaving reasonably well and most examples induce a gradient with smaller $r$, we might expect an example with a relatively higher difficulty to

be more acceptable. The case will be even more so with a decaying learning rate, which is common in practice.

### LAST

In the end, we present another explicit simple example that explicitly demonstrates the "easy-to-hard" intuition with a LAST teacher (Liu et al., 2021): for a linear regression learner with squared loss, a LAST teacher would synthesise the label

$$\tilde{y} = (1-\lambda)\langle w,x\rangle + \lambda\langle w^*,x\rangle,$$

where $\lambda = (\eta\langle x,x\rangle)^{-1}$. With a decaying learning rate, the focus of the teacher will gradually shifts from $\langle w,x\rangle$, the current prediction which is easier, to $\langle w^*,x\rangle$, the ground truth which is harder. This also provides some theoretical justification for the practice of self-training (Xie et al., 2020).

# Appendix B

# Theoretical Results

While we are able to demonstrate the effectiveness of various IMT methods proposed to some degree in empirical experiments, it will be ideal to have their performance guaranteed by theoretical results. Though this is an ambitious and challenging task in general, previous studies have shown some insightful results for specific cases (Liu et al., 2017, 2021; Qiu et al., 2022). In this section, we provide a high-level overview of these theoretical findings and their practical implications.

Since our focus will be on elucidating the core ideas and takeaways, we will not reiterate every theorem statement and proof here, but provide detailed pointers to sections in the original paper when appropriate.

## B.1 Overview

The majority of existing theoretical results are concerned with omniscient, greedy teaching. There are two types of results we will be mainly considering:

- **Universal speedup** results ask the question: "Can the IMT teacher always lead to faster convergence than SGD (so that not all the work we have done is in vain)?"
- **Exponential speedup** results study the *sample complexity*, i.e. the number of samples/iterations required to effectively teach the target model $w^*$ to a student.

  In fact, if we see IMT as an extension of MT (Zhu, 2013, 2015), sample complexity would be the IMT counterpart of teaching dimension. In (Liu et al., 2017), it is referred to as the *iterative teaching dimension*.

We note that exponential speedup is a stronger result than universal speedup, since SGD cannot achieve exponential speedup (Nemirovski et al., 2009).

In the following two sections, we discuss these two types of results respectively.

## B.2 Universal Speedup

Results on universal speedup have been provided for both joint teacher (Liu et al., 2017, Section 4.2) and LAST teacher (Liu et al., 2021, Section 3.2). Moreover, the LAST theorem can be easily transferred to DHT.

### B.2.1 Joint Teacher

**Definition 1** (Teaching Volume)**.** *For a specific student model, loss function $l$ and teaching space $\mathcal{X} \times \mathcal{Y}$, we define the teaching volume function (TV) at model parameter $w$ as*

$$TV(w) = \min_{x \in \mathcal{X}, y \in \mathcal{Y}} G(x, y|w)$$

Recall that $G$ is defined in Equation 2.4.

**Theorem 2** (Teaching Monotonicity)**.** *Given a student model, loss function $l$ and teaching space $\mathcal{X} \times \mathcal{Y}$, if for any $w_1, w_2$ that satisfy $\|w_1 - w^*\|^2 \leq \|w_2 - w^*\|^2$,*

$$\|w_1 - w^*\|^2 + TV(w_1) \leq \|w_2 - w^*\|^2 + TV(w_2)$$

*then with the same parameter initialisation and learning rate, the IMT teacher can always converge not slower than a random teacher.*

Intuitively, $TV(w)$ is the improvement in parameter discrepancy the teacher can and will achieve for a student at $w$ in one iteration, thus $\|w - w^*\|^2 + TV(w)$ is parameter discrepancy after the teacher works. Theorem 2 basically says that, as long as the teacher keeps a student who started up better ending up better (closer to $w^*$), it can achieve convergence no slower than SGD. The proof is straightforward using induction.

In practice, it is generally hard to show a set of teaching settings satisfies the monotonicity condition. Nonetheless, we give an example of a simple setting that suffices.

**Proposition 3.** *For a linear student $f(x) = \langle w, x \rangle$, square loss $l = \frac{1}{2}(f(x) - y)^2$ and teaching space $\mathcal{X} = \{x : \|x\| \leq R\}, \mathcal{Y} = \mathbb{R}$, the monotonicity condition is satisfied.*

### B.2.2 LAST

The universal speedup theorem for LAST follows the same idea as Theorem 2, except that everything is now in expectation term since LAST synthesises labels for a randomly sampled input. We do not reiterate the theorem here but refer to (Liu et al., 2021, Proposition 1).

## B.3 Exponential Speedup

Exponential speedup is also commonly referred to as exponential teachability, and have been studied for all joint synthesis teacher (Liu et al., 2017, Section 4.3), LAST (Liu et al., 2021, Section 3.4) and DHT (Qiu et al., 2022, Section 3.4). In general, an exponential speedup theorem takes the following form:

**Theorem 4** (Exponential Speedup Theorem Template)**.** *Given a specific teacher, student and algorithm, if the loss function satisfies certain conditions, then the teacher can guide the student to learn some $\varepsilon$-approximation of $w^*$ with $\mathcal{O}(C \cdot log\frac{1}{\varepsilon})$ samples, where $C$ is a constant that can depend on the teacher, student, algorithm and loss function.*

Different from the high probability bounds in traditional sample complexity, the sample complexity for IMT teachers is *deterministic* since a teacher uses a deterministic teaching strategy. Again, for LAST/DHT teachers, the stochasticity from random samples is encapsulated in expectations.

The key thing to note is that, in Theorem 4 (as well as Theorem 2 for universal speedup), we condition on the full teaching setting, including the teacher, student, algorithm and properties of the loss function[1]. Again, it is generally hard to prove a specific setting is exponentially teachable in practice, or to design a teacher that can achieve exponential teachability according to the theories. However, these theoretical results can offer us valuable insights in practice.

In the rest of this section, we will focus on discussing the high-level ideas and practical implications of these theorems, but omit the details of the exact mathematical statements.

[1] In contrast, the teaching dimension in MT only depends on the student model.

### B.3.1 Joint Teacher

In (Liu et al., 2017, Section 4.3), the authors studied the sample complexity for three types of joint synthesis teachers with different teaching capacities. In decreasing order:

- Synthesis-based teacher[2]: $\mathcal{X} = \{x : \|x\| \leq R\}$
- Combination-based teacher: $\mathcal{X} = \{x : \|x\| \leq R; x = \sum_i \alpha_i x_i, \alpha_i \in \mathbb{R}, x_i \in \mathcal{D}\}$
- Rescalable pool-based teacher: $\mathcal{X} = \{x : \|x\| \leq R; x = \gamma x_i, \gamma \in \mathbb{R}, x_i \in \mathcal{D}\}$

In all cases, $\mathcal{Y} = \mathbb{R}$ for regression task or $\{0,1\}$ for classification task.

Here, the radius $R$ can be interpreted as the knowledge domain of the teacher, and $\mathcal{D} = \{x_1, \ldots, x_m\}$ is a candidate pool.

In summary, the theoretical results say that:

- Given the setting, if exponential teachability can be achieved, a teacher with larger $R$ is more powerful, leading to faster convergence (smaller $C$ as in Theorem 4)
- A combination-based teacher can achieve the same performance as a synthesis-based teacher if the initial parameter displacement lies in the span of the candidate pool, i.e. $w^0 - w^* \in span(\mathcal{D})$.
- For a rescalable pool-based teacher, the richness of the pool decides whether exponential teachability can be achieved as well as the convergence speed, i.e. $C$. As the pool gets sufficiently rich, the rescalable pool-based teacher recovers a synthesis-based teacher.
- Combining the last two point, it is nice to have a rich pool.

These theoretical results largely agree with our intuition that a teacher with greater capability is more powerful, and agree with the empirical results in Section 2.4.

### B.3.2 LAST

For theoretical results on LAST, we mainly view LAST as a gradient rescaler (Section 2.2.2), say, $\nabla_w \tilde{l} = g(y) \cdot \nabla_w l$. The theorems then condition exponential teachability and convergence speed $C$ on $g(y)$, highlighting its importance. Also, it means that they can be applied to not only omniscient, greedy LAST, but also other types of LAST teacher.

[2]As named in the original paper.

It is interesting to note that one of the theorems (Liu et al., 2021, Theorem 2) does not actually involve the optimal parameter $w^*$, and so in principle can be applied to black-box LAST. However, it only applies to the case of scalar $y$. The more important point is that, when compared to a theorem that does utilise $w^*$ (Liu et al., 2021, Theorem 1), black-box teaching is indeed a harder problem than omniscient teaching.

The results for DHT are a direct extension of those for LAST (Qiu et al., 2022, Theorem 1); thus, we refrain from repeating them here.

## B.4 Summary

In this section, we review the existing theoretical results of IMT methods. Like many machine learning theories, while direct application in practice is still challenging, they have revealed intriguing insights and provide a robust foundation for our intuitions.

# Appendix C

# Experimental Details

This section contains details of the setup used in our experiments. When possible, we keep consistent with the setup used in previous works Liu et al. (2017, 2021); Qiu et al. (2022).

**Student Model** We consider a linear logistic regression (LR) learner with bias, without regularisation,

$$\min_{w\in\mathbb{R}^d, b\in\mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} l\left(\sigma(w^T x_i + b), y_i)\right),$$

where $\sigma$ is the sigmoid function, $l$ is the cross entropy loss.

We use the same student initialisation $w^0$ in all experiments for fair comparison. The optimal model $w^*$ is pre-trained from the same initialisation.

**Experiments on Half-moon** For the half-moon dataset, We use the build-in function from scikit-learn to generate 1000 (800/200) samples points with noise 0.2. We train the student model for 1000 iterations with learning rate 0.01.

**Experiments on MNIST** For the MNIST dataset, we use 1100 (1000/100) images from class 3 and 5. Following Liu et al. (2017, 2021); Qiu et al. (2022), we 24D random features projected by a fix random matrix $\mathbb{R}^{784\times 24}$. We train the student model for 1000 iterations with learning rate 0.0001.

**Parametrised Teacher** For parametrised teachers, we use a two-hidden layer MLP (input dimension - 128 - 128 - output dimension) with ReLU activations. The teacher takes input $x$, $y$, $w$ and $w - w^*$, where $w$ and $w^*$ are normalised.

In teacher training, the teacher is trained using a default Adam optimiser for 1000 iterations, unrolling 40 steps in each iteration. The student in each iteration is initialised with $w^0$ unless stated otherwise. We use teacher loss with the auxiliary term (Equation 3.3), with a weight decaying at rate 0.95.

# References


Agarwal, P. K., Har-Peled, S., Varadarajan, K. R., et al. (2005). Geometric approximation via coresets. *Combinatorial and computational geometry*, 52(1):1–30.

Alfeld, S., Zhu, X., and Barford, P. (2016). Data poisoning attacks against autoregressive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Alfeld, S., Zhu, X., and Barford, P. (2017). Explicit defense actions against test-set attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Bohdal, O., Yang, Y., and Hospedales, T. (2020). Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*.

Buciluǎ, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 535–541.

Campbell, T. and Broderick, T. (2018). Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, pages 698–706. PMLR.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Fan, Y., Tian, F., Qin, T., Li, X.-Y., and Liu, T.-Y. (2018). Learning to teach. *arXiv preprint arXiv:1805.03643*.

Fan, Y., Xia, Y., Wu, L., Xie, S., Liu, W., Bian, J., Qin, T., and Li, X.-Y. (2021). Learning to reweight with deep interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7385–7393.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Goldman, S. A. and Kearns, M. J. (1995). On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Har-Peled, S. and Kushal, A. (2005). Smaller coresets for k-median and k-means clustering. In *Proceedings of the twenty-first annual symposium on computational geometry*, pages 126–134.

Hardt, M., Jagadeesan, M., and Mendler-Dünner, C. (2022). Performative power. *Advances in Neural Information Processing Systems*, 35:22969–22981.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. (2016). Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Huggins, J., Campbell, T., and Broderick, T. (2016). Coresets for scalable bayesian logistic regression. *Advances in neural information processing systems*, 29.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Kumar, M., Packer, B., and Koller, D. (2010). Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23.

Lessard, L., Zhang, X., and Zhu, X. (2019). An optimal control approach to sequential machine teaching. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2495–2503. PMLR.

Liu, W., Dai, B., Humayun, A., Tay, C., Yu, C., Smith, L. B., Rehg, J. M., and Song, L. (2017). Iterative machine teaching. In *International Conference on Machine Learning*, pages 2149–2158. PMLR.

Liu, W., Dai, B., Li, X., Liu, Z., Rehg, J., and Song, L. (2018). Towards black-box iterative machine teaching. In *International Conference on Machine Learning*, pages 3141–3149. PMLR.

Liu, W., Liu, Z., Wang, H., Paull, L., Schölkopf, B., and Weller, A. (2021). Iterative teaching by label synthesis. *Advances in Neural Information Processing Systems*, 34:21681–21695.

Maclaurin, D., Duvenaud, D., and Adams, R. (2015). Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mei, S. and Zhu, X. (2015). Using machine teaching to identify optimal training-set attacks on machine learners. In *Proceedings of the aaai conference on artificial intelligence*, volume 29.

Mirzasoleiman, B., Bilmes, J., and Leskovec, J. (2020). Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609.

Peltola, T., Çelikok, M. M., Daee, P., and Kaski, S. (2019). Machine teaching of active sequential learners. *Advances in neural information processing systems*, 32.

Perdomo, J., Zrnic, T., Mendler-Dünner, C., and Hardt, M. (2020). Performative prediction. In *International Conference on Machine Learning*, pages 7599–7609. PMLR.

Qiu, Z., Liu, W., Xiao, T. Z., Liu, Z., Bhatt, U., Luo, Y., Weller, A., and Schölkopf, B. (2022). Iterative teaching by data hallucination. *arXiv preprint arXiv:2210.17467*.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Sener, O. and Savarese, S. (2017). Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Settles, B. (2009). Active learning literature survey.

Shinohara, A. and Miyano, S. (1991). Teachability in computational learning. *New Generation Computing*, 8:337–347.

Such, F. P., Rawal, A., Lehman, J., Stanley, K., and Clune, J. (2020). Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *International Conference on Machine Learning*, pages 9206–9216. PMLR.

Sucholutsky, I. and Schonlau, M. (2021). Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Tsang, I. W., Kwok, J. T., Cheung, P.-M., and Cristianini, N. (2005). Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6(4).

Wang, J., Wang, P., and Shafto, P. (2020a). Sequential cooperative bayesian inference. In *International Conference on Machine Learning*, pages 10039–10049. PMLR.

Wang, P., Paranamana, P., and Shafto, P. (2019). Generalizing the theory of cooperative inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1841–1850. PMLR.

Wang, P., Wang, J., Paranamana, P., and Shafto, P. (2020b). A mathematical theory of cooperative communication. *Advances in Neural Information Processing Systems*, 33:17582–17593.

Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A. (2018). Dataset distillation. *arXiv preprint arXiv:1811.10959*.

Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural networks*, 1(4):339–356.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Wu, L., Tian, F., Xia, Y., Fan, Y., Qin, T., Jian-Huang, L., and Liu, T.-Y. (2018). Learning to teach with dynamic loss functions. *Advances in neural information processing systems*, 31.

Xie, L., Wang, J., Wei, Z., Wang, M., and Tian, Q. (2016). Disturblabel: Regularizing CNN on the loss layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4753–4762.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698.

Xu, Z., Chen, B., Li, C., Liu, W., Song, L., Lin, Y., and Shrivastava, A. (2021). Locality sensitive teaching. *Advances in Neural Information Processing Systems*, 34:18049–18062.

Yang, S. C.-H., Yu, Y., Wang, P., Vong, W. K., Shafto, P., et al. (2018). Optimal cooperative inference. In *International Conference on Artificial Intelligence and Statistics*, pages 376–385. PMLR.

Yuan, L., Zhou, D., Shen, J., Gao, J., Chen, J. L., Gu, Q., Wu, Y. N., and Zhu, S.-C. (2021). Iterative teacher-aware learning. *Advances in Neural Information Processing Systems*, 34:29231–29245.

Zhang, C., Cao, X., Liu, W., Tsang, I., and Kwok, J. (2023). Nonparametric iterative machine teaching. *arXiv preprint arXiv:2306.03007*.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhao, B. and Bilen, H. (2021). Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR.

Zhao, B., Mopuri, K. R., and Bilen, H. (2020). Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*.

Zhu, J. (2013). Machine teaching for bayesian learners in the exponential family. *Advances in Neural Information Processing Systems*, 26.

Zhu, X. (2015). Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Zhu, X., Singla, A., Zilles, S., and Rafferty, A. N. (2018). An overview of machine teaching. *arXiv preprint arXiv:1801.05927*.