Learning with the Help of a Teacher

By

Ayon Sen

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
(Computer Sciences)

at the
University of Wisconsin-Madison
2020

Date of final oral examination: 06/15/2020

This dissertation is approved by the following members of the Final Oral Committee:

Xiaojin Zhu, Professor, Computer Sciences
Yingyu Liang, Assistant Professor, Computer Sciences
Robert Nowak, Professor, Engineering
Martina Rau, Associate Professor, Educational Psychology

I would like to dedicate this thesis to my loving parents . . .

Abstract

Machine learning has been playing an important role in our lives for quite a while now. It has been used for a multitude of applications ranging from voice recognition to disease detection. In traditional machine learning, a learning algorithm is presented with a large set of data and it is the task of the algorithm to find a model that explains the data best. This process, even though useful, can be very time consuming. With the help of a teacher the time required to train a learner can be reduced drastically. Training a machine learner with the help of a teacher is known as optimal control for machine learning (also known as machine teaching), and is applicable in a multitude of domains including education and security. In this thesis, we will explore some such usefulness of optimal control. We first provide evidence that optimal control can be helpful to teach humans, both for learning perceptual fluency in chemistry and the pronunciation of written words. Our research also verifies that an artificial neural network can be useful as a cognitive model for humans. Then we present a new type of attack on machine learners called training set camouflage. In this attack a malicious agent can train a publicly available learner on a sensitive task with the help of a benign looking training set. The malicious agent does not raise any suspicion while passing this training set. Finally, we show a comparative analysis of the different metrics used in adversarial machine learning. In particular, through human experiments we show that even though none of the metrics currently used in practice is suitable, pixel 3-norm provides the best approximation. The research works presented in this thesis show how a teacher can play an important and useful role to train a machine learning algorithm.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

The study of artificial intelligence (AI) is inspired by *natural intelligence* (NI) displayed by humans and other animals. In computer science, AI is defined as the study of intelligent agents. In particular, the broad goal of machine learning is to gain insight regarding human learning processes and then transferring them to agents known as machine learners. Understanding the human learning process can help both machine learners and humans. This is one of the avenues where machine teaching comes into focus.

Machine teaching [174] can be viewed as the inverse problem of machine learning. Machine learning refers to computer algorithms which find an optimal model given a set of independent and identically distributed (i.i.d.) data. Put another way, such algorithms find a model that fits or explains the data best. In machine teaching, the machine learning algorithm is a student who is guided by a teacher. The teacher wants the student to learn a target model and has knowledge about the learning algorithm, and teaches by giving the student training examples. Machine teaching wants to find an optimal training set (usually smallest) such that given this training set a learning algorithm learns a target model. Note that unlike traditional teaching, the teacher's role here is not necessarily benevolent. In an adversarial setting, the attacker (teacher) minimally poisons the training data to force the learner to learn a nefarious model.

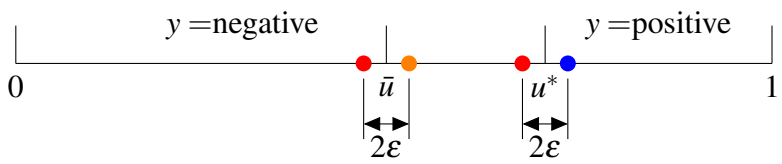


Fig. 1.1 Teaching a 1D threshold classifier. Red and blue dots represent negative and positive examples respectively. The orange dot represents a negative example whose label is incorrectly presented to the learner by an attacker.

Figure 1.1 shows an example of machine teaching in action. The goal is to learn a 1D threshold classifier where the input distribution P_U is uniform over the interval [0, 1]. The true threshold is at u^* and the binary label is noiseless: $y = \mathbb{1}(u \geq u^*)$. Here, $\mathbb{1}$ is the indicator function. In passive machine learning the learner receives a training set: $\{u_i\}_{i=1}^n \overset{i.i.d.}{\sim} P_u$ with the corresponding label $y_i = \mathbb{1}(u_i \geq u^*)$. It can be shown that if the learner is consistent, then to achieve a generalization error of ε the learner needs $n \geq O(n^{-1})$ training examples. It is easy to see that a teacher, on the other hand, can construct a teaching set by carefully picking two training examples, one positive and one negative centered at u^* such that they are at most 2ε apart. The learner trained on this set achieves generalization error of ε by definition, and importantly the size of the teaching set remains two irrespective of the magnitude of ε. In the adversarial setting, the attacker wants the learner to learn a threshold $\bar{u}$ different from u^*. The attacker can achieve her goal by picking two examples around $\bar{u}$ just like before but lying about the corresponding labels as necessary (the orange dot in the figure).

Machine teaching can also be formulated as optimal control for machine learners. Henceforth, we use machine teaching and optimal control for machine learning interchangeably. Optimal control [170] deals with finding a control law for a dynamical system over a period of time such that an objective function can be optimized. In control language the machine learning algorithm is the plant, the state is the model estimate, and the input is the training data. The teacher plays the role of the controller. Recently there has been a surge of interest in optimal control for machine learning with a diverse set of applications in the field of education [104, 171, 89], interactive machine learning [143], program synthesis [64] and security [4, 5, 90, 167] among others.

In particular, optimal control can have vast implications in the fields of education and security. For example, it can be used to teach humans to perform well on a particular task, devise an effective curricula for a class of diverse students and help create an adaptive syllabus based on student performance. Many of the intelligent tutoring systems (ITS) [152, 114] currently used in practice improve the student's experience by mainly identifying how much expert (mastery) a student has become on particular topics. After identifying topics they are weak on, the software then provides examples related to that topic. Such examples are pre-selected by a teacher and does not take the dynamics of learning directly into account. Optimal control on the other hand takes such dynamics into account. This has the potential of a smoother student's experience as teaching would be catered to her understanding.

Similarly optimal control has a lot of potential at identifying threats in the domain of machine learning security. Study of adversarial examples in the domain of machine learning has been a hot topic over the last few years. One of the main reasons behind the existence of adversarial examples stems from the difference between human and machine learning

procedures. This is why instances incapable of misguiding humans can still easily fool machine learners. For example, it was first showed by [145] that simple perturbations to images, which are imperceptible to human eyes, can cause a classification algorithm to misclassify. Optimal control can be used to identify how such a learner can be attacked and at the same time develop effective defensive strategies.

In this thesis, we explore how optimal control can be used to get a better understanding of human learning which in turn can benefit both humans and machine learners. First, we show how it can be used to help students learn perceptual fluency in the chemistry domain. We compared learning gains from a randomly generated training sequence, an expert generated training sequence and a machine teaching generated training sequence. Our experiments found that the machine generated training sequence helped weaker students to generalize better leading to higher learning gains. We then also applied a similar technique for learning phonological representation from orthographic representation i.e., learning to correctly pronounce words from written words. For this study we were able to improve generalization gains compared to multiple baselines where the words are sampled with respect to their natural frequency in the environment for a long training sequence. Then we show how optimal control can be used for a novel attack setting called training set camouflage. In this setting, a malicious agent Alice wants to train an off the shelf learner Bob on some secret task by sending him a training set. The message channel is observed by an eavesdropper Eve who can stop the communication between Alice and Bob if she suspects Alice's intent is not noble. Thus to fool Eve, Alice constructs a special training set generated from a publicly available innocuous dataset. This dataset seems benign to Eve but when used for training can make Bob learn a model which is approximately good on the original class. Finally, we show a comparative analysis of different popular metrics used in adversarial machine learning. In such a setting an adversary slightly perturbs a natural image so that the difference is not noticeable to human, however the labels of the perturbed image is changed according to a trained classifier. We show that none of the metrics used in adversarial literature is suitable by using the notion of just noticeable difference (JND). Our study hopes to shed a better light to the metrics used in practice. The rest of this thesis is organized as follows. In Chapter 2 we present the research works related to optimal control in machine learning, education and security. In Chapter 3 - 6 we discuss the successful uses of machine teaching.

Chapter 2

Related Work

In this chapter we review the related research in the fields of machine teaching, learning with visual representations, reading development, security and adversarial machine learning in the image domain.

2.1 Machine Teaching

As mentioned previously, machine teaching is the inverse of machine learning and has applications in various fields [168, 84]. The general idea of machine teaching is to construct a training dataset keeping a particular student (a machine learning algorithm) and a target model in mind. In recent years machine teaching has seen a surge in interest. It has been applied in multiple fields ranging from human teaching to computer security. A general machine teaching framework is proposed by Zhu [172]. A variety of different approaches have been explored to date for modeling the teaching of students. Goldman & Kearns [49] studied the complexity of teaching and introduced the term teaching dimension as the minimum number of instances a teacher must show to the student to uniquely identify any hypothesis from the hypothesis space. Liu & Zhu [84] provide the teaching dimension of several linear learners. Zhu [171] considers Bayesian learner in exponential family as the student model, and formulates machine teaching as an optimization problem over teaching examples that balance the future loss of the learner and the effort of the teacher. The application of machine teaching in a heterogeneous class setting was explored in [173]. They propose a minimax teaching criterion to guarantee the performance of the worst learner in the class. Machine teaching has also been explored in interactive settings. Cakmak & Thomaz [23] considered the scenario where a human teacher helps a student perform a classification task by showing examples. Their main focus was to improve the human teacher by giving teaching guidance. Singla et al. of [139] consider the scenario of training crowd source workers. Such workers

are not always reliable and are prone to mistakes that an expert will not make. The authors propose a sequential teaching algorithm that can teach the crowd workers on a given task first. Afterwards they are shown to have actually performed much reliably on the task. However, their work does not consider a particular cognitive model of the worker. Liu et al. [86] assumes stochastic gradient descent learners with a different teacher and student representation space. They propose a greedy approach to solve the optimization problem. They extended their work to the case where the hyperparameters of the learner is not known in advance [85]. In [89] Ma et al. show how a teacher can trim down an independently and identically distributed training set while leading to a higher gain for the learner. This is known as super teaching. Machine teaching can also be used to identify misleading examples and historical biases in the training set. Removing these examples lead to better learning. This is known as training set debugging [168].

Machine teaching also has applications in the domain of adversarial learning which studies the use of machine learning in security-sensitive domains. Numerous attacks against various machine learners have been explored, highlighting the security ramifications of using machine learning in practice [60, 82]. Barreno et al. [11, 10] present a taxonomy of identifying and analyzing attacks against machine learning systems. One important aspect of analyzing security threats is the cost of attacker and defender. The authors provide a formal structure of how the such costs impact both parties. They also provide some defense mechanism against such attack. Dalvi et al. [31] formulate classification as a game between the attacker and defender (learner) and propose a strategy under which the learner is optimal given the adversary's optimal strategy. Tan, Killourhy & Maxion [146] show how an adversary can craft an offensive mechanism that renders an anomaly-based intrusion detector blind to the presence of common attacks. More recently, it was shown that data poisoning attacks can also be effective against differentially private learners [91]. Such attacks can also work against batch reinforcement learning algorithms [90].

2.2 Learning with Visual Representation

Theories of learning with visual representations define visual representations as a specific type of external representation. External representations are objects that stand for something other than themselves — a referent [105]. When we see an image of a pizza, for example, the referent could be a slice of pizza (a concrete object). Alternatively, when used in the context of math instruction, the referent could be a fraction of a whole pizza (an abstract concept). Representations used in instructional materials are defined as external representations because they are external to the viewer. By contrast, internal representations are mental objects that

students can imagine and mentally manipulate. Internal representations are the building blocks of mental models; these models constitute students' content knowledge of a particular topic or domain. External representations can be symbolic or visual. For instance, text or equations are symbolic external representations that consist of symbols that have arbitrary (or convention-based) mappings to the referent [128]. By contrast, *visual representations* have similarity-based mappings to the referent [128].

Several theories describe how students learn from visual representations. Mayer's [95] Cognitive Theory of Multimedia Learning (CTML) and Schnotz's [128] Integrated Model of Text and Picture Comprehension (ITPC) draw on information processing theory [8] to describe learning from external representations as the integration of new information into a mental model of the domain knowledge. Here, we focus on learning processes relevant to visual representations.

First, students select relevant *sensory information* from the visual representations for further processing in working memory. To this end, students use perceptual processes that capture visuo-spatial patterns of the representation in working memory [128]. To willfully direct their attention to relevant visual features, students draw on conceptual competencies that enable top-down thematic selection of visual features [1, 55].

Second, students *organize* this information into an internal representation that describes or depicts the information presented in the external representation. Because visual representations have similarity-based analog mappings to referents, their structure can be directly mapped to the analog internal representations [44, 128]. In forming the internal representation, students engage perceptual processes that draw on pattern recognition of objects based on visual cues. They engage conceptual processes to map the visual cues to conceptual representational competencies that allow the retrieval of concepts associated with these objects. The resulting internal representation is a perceptual analog of the visual representation. It is depictive in that its organization directly corresponds to the visuo-spatial organization of the external visual representation [128].

Third, students integrate the information contained in the internal representations into a *mental model* of the domain knowledge (e.g., schemas, category knowledge). To this end, students integrate the analog internal representation into a mental model by mapping the analog features to information in long-term memory. This third step is what constitutes learning: students learn by integrating internal representations into a coherent mental model of the domain knowledge [95, 128, 161].

In sum, students' learning from visual representations hinges on their ability to form accurate internal representations of the representations' referents and on their ability to integrate internal representations into a coherent mental model of the domain knowledge. This

process involves conceptual representational competencies as well as perceptual fluency [109]. Although it is well established that conceptual and perceptual competencies are interrelated [51, 55], it makes sense to distinguish them because they are acquired via qualitatively different learning processes [51, 68, 72]. As mentioned earlier, conceptual representational competencies are acquired via verbally mediated, explicit processes [72, 109]. By contrast, perceptual fluency is acquired via implicit, mostly nonverbal processes. Whereas most prior research on instructional interventions for representational competencies has focused on conceptual processes, we focus on perceptual processes.

Research on perceptual fluency is based on findings that experts can automatically see meaningful connections among representations, that it takes them little cognitive effort to translate among representations, and that they can quickly and effortlessly integrate information distributed across representations [47]. Such perceptual fluency frees cognitive resources for explanation-based reasoning [50, 119] and is considered an important goal in STEM education. According to the CTML and the ITCP, perceptual fluency involves efficient formation of accurate internal representations of visual representations [95, 128]. Perceptual fluency also involves the ability to combine information from different visual representations without any perceived mental effort and to quickly translate among them [27] [68]. According to the CTML and ITCP, this allows students to map analog internal representations of multiple visual representations to one another [95, 128].

Cognitive science literature [47, 1, 72] suggests that students acquire perceptual fluency via perceptual-induction processes. These processes are inductive because students can infer how visual features map to concepts through experience with many examples [47, 1, 68]. Students gain *efficiency* in seeing meaning in visuals via perceptual chunking. Rather than mapping specific analog features to concepts, students learn to treat each analog visual feature as one perceptual chunk that relates to multiple concepts. Perceptual-induction processes are thought to be nonverbal because they do not require explicit reasoning [72]. They are implicit because they occur unintentionally and sometimes unconsciously [132].

Interventions that target perceptual fluency are relatively novel. Kellman and colleagues [68] developed interventions that engage students in perceptual-induction processes by exposing them to many short problems where they have to rapidly translate between representations. For example, students might receive numerous problems that ask them to judge whether two visuals show the same item. These interventions have enhanced students' learning in domains like chemistry [113, 157]. Perceptual learning is strongly affected by problem sequences [109]. To design appropriate problem sequences, consecutive problems expose students to systematic variation (often in the form of contrasting cases) so that irrelevant features vary but relevant features appear across several problems [68].

2.3 Reading Development

Early reading development rests on the ability of the child to learn as much as possible about how print and speech are related in order to move on to the more important, subsequent aspects of reading, namely comprehension [129]. This creates an important issue for education: How can early reading experiences be structured to support speedy development, including the capacity to generalize [28]? The prevailing theory embodied in commercial reading curricula assumes that learners need to start with the smallest words that contain the most predictable spoken patterns, growing over time to be taught about longer words and larger, less predictable patterns of print. Programs of instruction vary in their approach, in part because no comprehensive theory of how to structure the learner's print environment over time exists. There are a variety of theories concerning which aspects of sublexical structure children learn to read, and in what order [154], though no theory of teaching exists in this domain that deals with the corresponding teaching problem. This is not suprising given its combinatorial complexity. Making optimal choices about what to introduce into the child's experience and when requires experimentation over high dimensional aspects of language structure and how that structure can be introduced over time. The experimentation reported on here adds to this literature by casting early reading development as an optimization problem. Our approach seeks to understand how the learner's experience can be structured in such a way to learn as much as possible in a limited timeframe so that the child can learn quickly, and generalize to many untaught forms as possible.

2.4 Security

With the widespread use of digital technology it is now easier than before to gather information while sitting home. This also raises potential questions regarding security of such information. The question of such security has been explored for a very long time. The advent of technology and the ability of computers to process millions of functionalities in a very short time has added many more layers to it. There are multiple branches of security research like in the fields of steganography and cryptography. We discuss various research works related to security in this section.

Concealing the existence of messages is known as steganography. One illustration of steganography (first presented in 1983 in [137]) would be two prisoners Alice and Bob who wish to devise an escape plan. All their communication is observed by the adversary (the warden, Eve) who will thwart their plan as soon as she detects any sign of hidden message. In this setup, the mode of communication is text. An information-theoretic model for such

a setup is presented in [22]. Steganography has multiple useful applications [165, 65]. Although many different data formats can be used for steganography, images [108, 65] are by far the most popular format due to their popularity on the internet. Image steganography can be broadly classified into spatial domain, transform domain, spread spectrum and model based. In spatial domain steganography, embedding the secret message is done on the image pixels directly [65, 159]. For transform domain steganography, embedding the message requires transforming the image from the spatial domain to the frequency domain by using some transformation like discrete cosine transform (DCT) [13], discrete wavelet transform (DWT) [54], singular value decomposition (SVD) [58] etc. The spread spectrum steganography techniques [94, 149] are characterized by their robustness against statistical attacks. Model based steganography [126] only alters part of the cover image to carry the secret message without modifying the cover statistical properties. This method is relatively new and is also known as adaptive steganography. Steganalysis is the study of detecting the existence of hidden messages (using steganography). A study of steganography from a complexity-theoretic point of view was presented in [59, 117]. This complexity-theoretic security notion is similar to modern cryptography and they try to define a secure stegosystem such that the stegotext is computationally indistinguishable from the covertext. In such a scenario a new term called steganographic secrecy of stegosystem is introduced which is defined as the inability of a polynomial-time adversary (Eve) to distinguish between observed distributions of unaltered covertext and stegotexts. To the best of our knowledge steganographic techniques have not been used in the domain of training sets for machine learning models.

Steganography is often confused with cryptography [67, 151] even though the goal of these two systems are completely different. The goal of cryptography is to ensure confidentiality of data in communication and storage processes. Hiding the existence of sensitive data is not the end goal here (unlike steganography). According to Kerckoff's principle [69, 70], this confidentiality must not rely on the obfuscation of the encoding scheme, but only on the secrecy of the decryption key. Encryption schemes can be broadly categorized in two ways: symmetric and asymmetric. In the symmetric scheme [100, 30, 153, 37] both Alice and Bob share the encryption/decryption key. The main advantage of the symmetric schemes is fast computation. But its application is limited because Alice and Bob have to determine the key beforehand and also because a separate key is required for interaction between different parties. Asymmetric schemes [121, 38], on the other hand, use a separate encryption and decryption key. In this scenario Alice uses her *public key* to encrypt the data while Bob uses his *private key* to decrypt it. The main advantage of this scheme is that the sender and receiver do not have to agree on anything before passing the the data

and hence these are widely used in practice. The main disadvantage of this scheme is high computation time.

There is another branch of cryptography which deals with performing computation on encrypted data. One usefulness of such a scheme would be when we delegate some computation to an untrusted computer. The untrusted computer will perform the computation on encrypted data, hence will not know the real contents of it, and return the computed value. For coherence, the result sent back by the untrusted computer has to equal to the intended computed value if performed on the original data. This scheme is known as *homomorphic encryption* and was first proposed by Rivest et al. in 1978 [120]. A homomorphic cryptosystem which supports arbitrary computation on ciphertexts is known as fully homomorphic encryption (FHE). The first plausible construction of such a system was proposed in [140]. This scheme supports both addition and multiplication operations on ciphertexts, which in turn makes possible to construct circuits for arbitrary computations. Some second generation solutions were proposed in [18, 17, 87, 45]. These second generation cryptosystems feature much slower growth of the noise during the homomorphic computations.

2.5 Adversarial Machine Learning in the Image Domain

The existence of adversarial examples in the image domain was first brought to light by the research presented in [145]. Given a natural image $\mathbf{x}_0$ the authors find a modified image $\mathbf{x}$ such that $||\mathbf{x}-\mathbf{x}_0||_2 < \varepsilon$. They suggest that for a small ε, the changes to $\mathbf{x}_0$ will be imperceptible to humans. In the follow up work [52], the authors suggest that DNNs are susceptible to adversarial attacks due to their linear nature. They also propose the popular Fast Gradient Sign Method (FGSM) in the same paper which uses the ∞-norm as a measure of perturbation. The authors also show that for some examples the adversarial changes are imperceptible to human eyes. Following the discovery of adversarial examples, other researchers explored different avenues to generate adversarial examples. A study of the existence of adversarial examples in the physical world is presented by Kurakin, Goodfellow & Bengio [80]. The authors use the ∞-norm as their metric as they were building upon the work of Goodfellow, Shlens & Szegedy [52]. Another popular attack strategy DeepFool was proposed by Moosavi-Dezfooli, Fawzi & Frossard [99]. This is an iterative attack which uses both both $2-$ and $\infty-$norm to find perturbations that are imperceptible. At the same time other researchers also became interested in defending classifiers against such attacks. Projected gradient descent (PGD) using ∞-norm was proposed as the universal first-order adversary by Madry et al. [93]. They also claim that adversarial examples generated by PGD are imperceptible to humans and then devise strategies to defend DNNs against such attacks.

Another such work by Carlini & Wagner [25] evaluates the robustness of DNNs. They consider both $0-, 2-$ and $\infty-$norm perturbations as they are widely used in the adversarial attack literature. The same authors also introduce universal perturbations (which are again imperceptible to humans) in [98] using only the $2-$ and $\infty-$norms. They also state that their approach can be generalized for any p-norm metric. Using only 2-norm to generate adversarial examples was explored by both Sarkar et al. [127] and Baluja & Fischer [9]. On the other hand, Papernot et al. [103] and Su, Vargas & Sakurai [142] use 0-norm while generating adversarial examples. They use 0-norm as they want to minimize the number of pixels that get perturbed. In this regard the work presented by Su, Vargas & Sakurai is more interesting as they show that adversarial examples can be generated by modifying only one pixel.

There are a few research work in this domain which do not use pixel p-norm. Xiao et al. [162] first spatially transform the original image. They then apply the flow function to generate perturbation on this transformed domain. The authors claim that the resultant geometric changes in the pixel space are both small and locally smooth. Three different metrics inspired by computer vision techniques are proposed by Jang, Wu and Jha [63]. These three metrics are number of edges, Fast Fourier Transform and histogram of oriented gradients. In [169], Zhang, Zhu & Lessard use 2-norm, but on the DNN representation space. They use 2-norm as it is the most natural metric for measuring distance. Engstrom et al. [40] investigate the robustness of DNNs under natural image transformations. They show that a rotation followed by a translation is enough to fool DNNs.

Chapter 3

Teaching Perceptual Fluency to Humans

Visual representations are instructional tools that are used in science, technology, engineering and math (STEM) domains [3, 102]. One such example for teaching bonding in chemistry instruction is shown in Figure 3.1. We typically assume that such visuals help students learn as they make abstract concepts more accessible. But at the same time it is possible that such visual representations may impede students' learning if students do not know how the visuals show information [2]. For example, a chemistry student needs to learn that the dots in the Lewis structure in Figure 3.1(a) show electrons and that the spheres in the space-filling model in Figure 3.1(b) show regions where electrons likely reside.

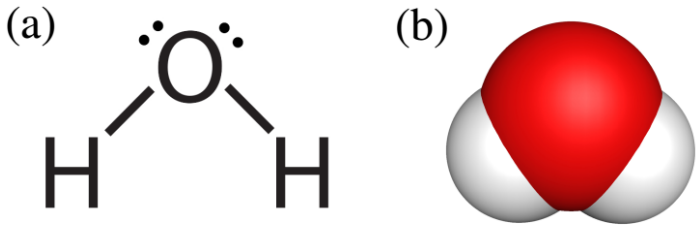


Fig. 3.1 Two commonly used visual representations of water (a: Lewis structure; b: space-filling model).

Instructional interventions that help students acquire representational competencies focus on *conceptual* representational competencies. These include the ability to map visual features to concepts, support conceptual reasoning with visuals, and choose appropriate visuals to illustrate a given concept [14]. For example, chemists can explain how the number of lines and dots shown in the Lewis structure relate to the colored spheres in the space-filling model by relating these visual features to chemical bonding concepts. Such conceptual

representational competencies are acquired via explicit, verbally mediated learning processes that are best supported by prompting students to explain how visuals show concepts [72, 109].

Less research has focused on a second type of representational competency — *perceptual fluency*. It involves the ability to rapidly and effortlessly see meaningful information in visual representations [47, 50]. For example, chemists immediately see that both visuals in Figure 3.1 show water without having to effortfully think about what the visual shows. They are as fluent at seeing meaning in multiple visuals as bilinguals are fluent in hearing meaning in multiple languages. Perceptual fluency frees up cognitive resources for higher-order complex reasoning, thereby allowing students to use visuals to learn new domain knowledge [51, 109].

Students acquire perceptual fluency via implicit inductive processes [47, 50]. These processes are nonverbal because verbal reasoning is not necessary [68] and may even interfere with the acquisition of perceptual fluency [72]. Consequently, instructional problems that enhance perceptual fluency engage students in simple problems to quickly judge what a visual shows [68]. For example, one type of perceptual-fluency problem may ask students to quickly and intuitively judge whether two visuals like the ones in Figure 3.1 show the same molecule. They ask students to rely on implicit intuitions when responding to a series of perceptual-fluency problems. Students typically receive numerous perceptual-fluency problems in a row. The problem sequence is typically chosen so that (1) students are exposed to a variety of visuals and (2) consecutive visuals vary incidental features while drawing students' attention to conceptually relevant features [68, 109].

However, these general principles are underspecified in the sense that they leave room for many possible problem sequences. To date, we lack a principled approach capable of identifying sequences of visual representations that yield optimal learning outcomes for perceptual-fluency problems. To address this issue, we developed a novel educational data mining approach. Using data from human students who learned with perceptual-fluency problems, we trained a machine learning algorithm to mimic human perceptual learning. Then, we used an algorithm to search over possible sequences of visual representations to identify the sequence that was most effective for a machine learning algorithm. In a human experiment, we then tested whether (1) the machine-selected sequence of visual representations yielded higher learning outcomes compared to (2) a random sequence and (3) a sequence generated by a human expert based on perceptual learning principles.

In the following, we describe the methods we used to identify the machine-selected sequence and the methods for the human experiment. We also discuss how our results may guide educational interventions for representational competencies and educational data mining more broadly.

3.1 Cognitive Model

For the application of machine teaching in this education domain we first need a cognitive model, or a learning algorithm ($\mathscr{A}$) that mimics how human students learn a particular problem. Given this cognitive model, machine teaching then seeks for a sequence of instances (optimal training sequence $\mathscr{O}$) such that given $\mathscr{O}$, the learning algorithm learns to solve the particular problem.

To evaluate whether a training sequence is effective, we test the cognitive model's performance at mapping visual representations using a different set of perceptual-fluency instances than used during training. Typically, a sequence of training instances (aka training problems in educational domain) is drawn from a distribution of perceptual-fluency instances used for training (P_t). The set of test instances comes from a separate distribution of perceptual-fluency instances (P_e). The goal is to minimize the test error rate on P_e. The goal of machine teaching then becomes:

$$\mathscr{O} = \underset{S \in \mathscr{C}_t}{\operatorname{argmin}} P_{(x,y)\sim P_e}\left(\mathscr{A}(S)(x) \neq y\right) \tag{3.1}$$

Here, $\mathscr{C}_t$ is the set of all possible training sequences and $\mathscr{A}(S)$ is the learned hypothesis after training on the sequence S. Note that, $\mathscr{O}$ is not necessarily an i.i.d. sequence drawn from P_t. To properly construct the optimal training sequence in this given setting, we must understand

1. the nature of the to-be-learned domain knowledge
2. the learning algorithm the cognitive model is using

In this work, the to-be-learned domain knowledge is well-known. It is the mappings between visual representations that students have to learn. Further, we used data from human students learning from perceptual-fluency instances to generate a cognitive model that mimics how humans learn mappings between visual representations. We now describe how we constructed the cognitive model that was used to construct the training sequence. To this end, we first describe the perceptual-fluency instances, then describe how we formally represented these instances (i.e., the feature vectors), which learning algorithm the cognitive model used, and finally how we used the cognitive model to identify the optimal training sequence.

3.1.1 Perceptual-Fluency Instances

Perceptual-fluency instances are single-step problems that ask students to make simple perceptual judgments. In our case, students were asked to judge whether two visual represen-

tations showed the same molecule, as shown in Figure 3.2. Students were given two images. One image was of a molecule represented by a Lewis structure and the other image was a molecule represented by a space-filling model. Their problem was to judge whether those two images show the same molecule or not.

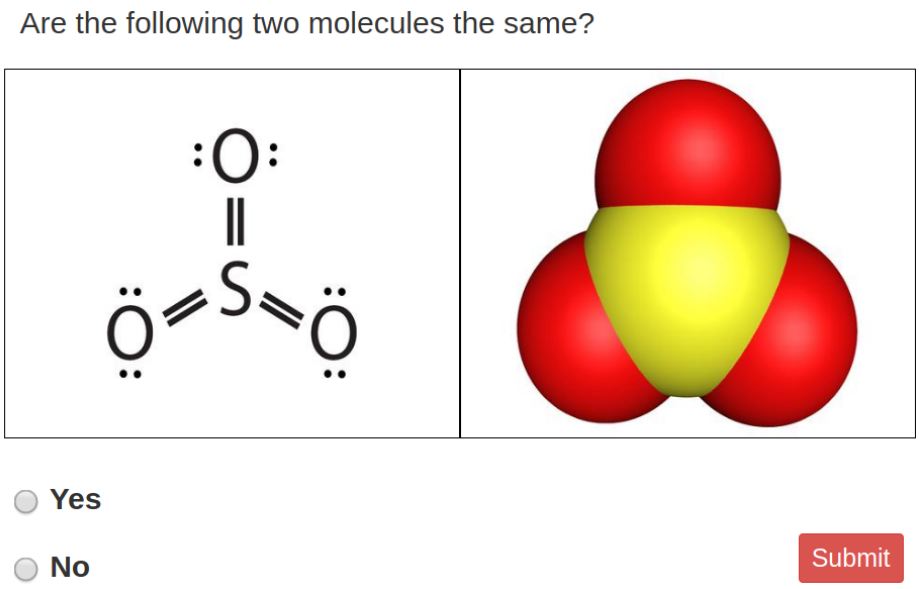


Fig. 3.2 In this sample perceptual-fluency instance, students judged whether or not the Lewis structure and the space-filling model showed the same molecule. The answer is yes.

3.1.2 Visual Representation of Molecules

In our experiment, we used visual representations of chemical molecules common in undergraduate instruction. To identify these molecules, we reviewed textbooks and web-based instructional materials. We counted the frequency of different molecules using their chemical names (e.g., H_2O) and common names (e.g., water), and chose the 142 most common molecules. In order to formally describe the visual representations, we quantified visual features for each of the molecules. To this end, we first hand-coded the visual features that were present in the visual representations. For Lewis structures, these hand-coded features included counts of individual letters as well as information about different bonds present in each molecule, among others. For space-filling models, hand-coded features included counts of colored spheres, bonds, and other features. Further, we included several surface features that we expect human students attend to based on findings that humans tend to focus on broader surface features that are easily perceivable. Then we used the method found in [112] to determine which subset of features (each for Lewis structure and space-filling

model) humans attend to most. Building on these results, we created feature vectors for each of the molecules (Figure 3.3). These feature vectors of Lewis structures and space-filling models contained 27 and 24 features, respectively. These feature vectors were then used to train and test the learning algorithm.

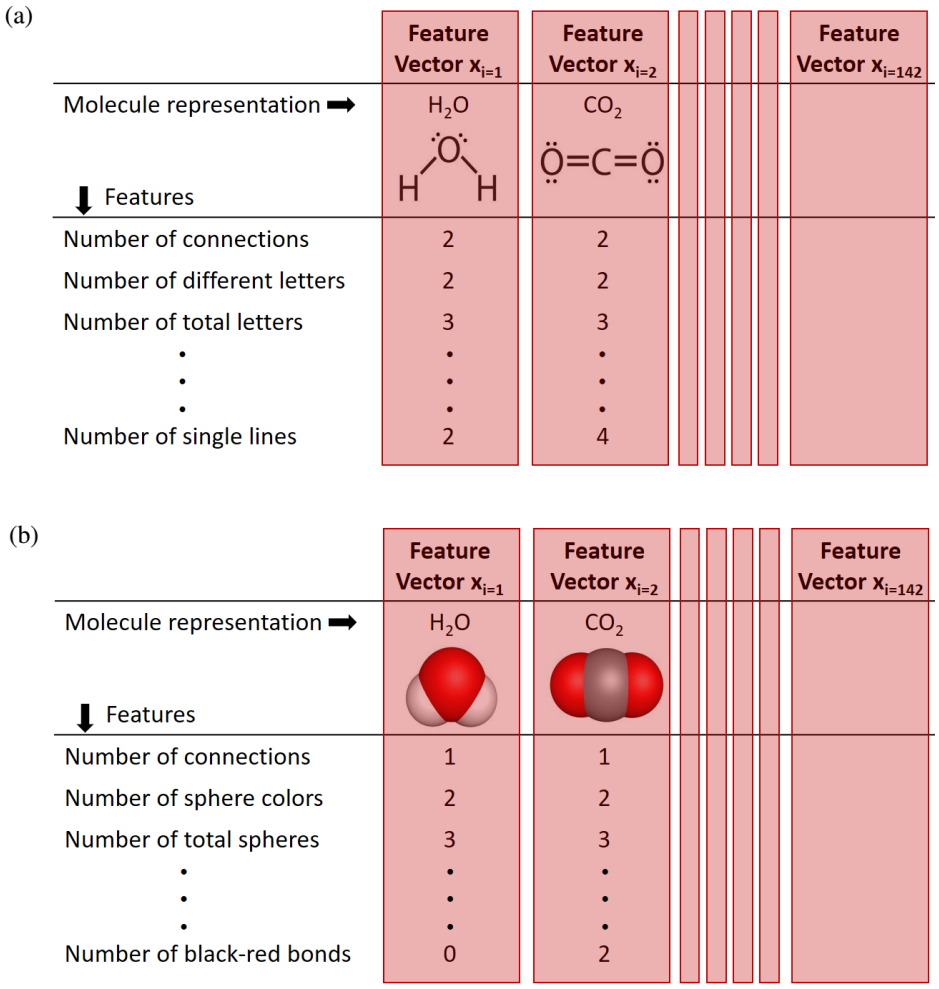


Fig. 3.3 Example features for H_2O and CO_2 molecule representations with feature vectors in red (a: Lewis structure; b: space-filling model).

3.1.3 Learning Algorithm

We used a feed-forward artificial neural network (ANN) [33] as our learning algorithm. The ANN took two feature vectors (x_1 and x_2) as input. Each feature vector corresponded to one of the two molecules shown. Given this input, the ANN produced a probability that the two molecules were the same. Then, given the correct answer $y \in \{0,1\}$ (here 1 means the two

molecules are the same), the ANN updated its weights using the backpropagation algorithm. The backpropagation algorithm uses gradients to converge to an optima. Algorithm 1 shows the training procedure of the neural network. It shows that the update procedure also used a history window and multiple backpropagation passes, an atypical approach for an ANN. Unlike an ANN learning algorithm, a human can learn from memory. Hence, we assumed that humans remember a fixed number of past consecutive instances. Further, we assumed that after receiving feedback on the latest instance, humans update their internal model by reviewing memorized instances (along with the latest instance) several times. To emulate this behavior, we introduced the history window and multiple backpropagation passes. This procedure was followed for all instances in a given training sequence.

Algorithm 1 train: training method for the NN learner

```
1: Input: Training sequence S , Learning rate η, History window size w, Number of
   backpropagations b
2: H ← [] //initialize an empty history window
3: for i = 1 → |S| do
4:    append(H, S[i]) //update history window
5:    // train on the history window
6:    w' ← |H|
7:    for k = 1 → b do
8:       for j = 1 → w' do
9:          (x, y) ← H[j]
10:         backprop(x, y, η)
11:      end for
12:   end for
13:   //check history window size
14:   if w' > w then
15:      H.remove(0) //remove the oldest instance in history
16:   end if
17: end for
```

Structurally, our learning algorithm also differed from a general artificial neural network in that it had two separate weight columns (one for each representation of the input molecules). The model architecture of the ANN is shown in Figure 3.4. Here the weights and outputs from one of the columns did not interact with those of the other column until the output layer. The network mapped the two inputs (feature vectors x_1 and x_2) to a space wherein the same molecule shown by different representations are close to each other while different molecules are distant. These mapping functions are called the embedding functions (one for each representation) and the space is called a common embedding space. Once the mapping was complete, a judgment was possible regarding the similarity of the input molecules. This

judgment was based on the distance in the common embedding space and made in the output layer of the ANN. Embeddings were generated in the layer before the output layer—the embedding layer.

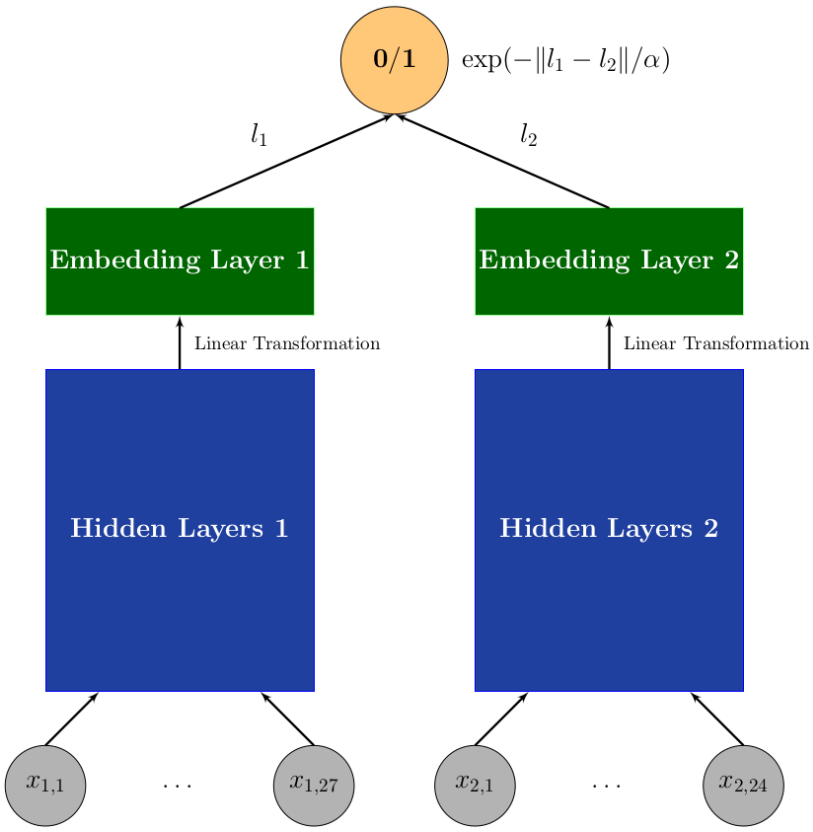


Fig. 3.4 Structure of the Artificial Neural Network learning algorithm

Neurons in an ANN use a non-linear function called activation function to introduce non-linearity. For all hidden layers before the embedding layer, we used the leaky rectifier [163] activation function (the neuron employing leaky rectifier is called a leaky rectified linear unit or leaky ReLU). A standard rectified linear unit (ReLU) allows only positive inputs to move onwards (outputs 0 otherwise). A leaky ReLU, on the other hand, outputs a small scaled input when the input is negative. Both ReLU and leaky ReLU have strong biological motivations. According to cognitive neuroscience studies of human brains, neurons encode information in a sparse and distributed fashion [7]. Using ReLU, ANNs can also encode information sparsely. Besides this biological plausibility, sparsity also confers mathematical benefits like information disentangling and linear separability. Rectified linear units also enable better training of ANNs [48]. The embedding layers, by contrast, do not use activation functions. Hence, the output of embedding layers are a linear transformation of its inputs.

Given the inputs (x_1, x_2), let the ANN-generated embeddings be l_1 and l_2, respectively. Then, we computed the probability of the two representations showing the same molecule in the output layer using the following equation:

$$\exp\left(-\frac{\|l_1 - l_2\|}{\alpha}\right) \tag{3.2}$$

Here, α is a trainable parameter that the ANN learns along with the weights. We thresholded this value at 0.5 to generate the ANN prediction $\hat{y} \in \{0, 1\}$.

Pilot Study - Train the Learning Algorithm

Our first step was to train the learning algorithm to mimic human perceptual learning. To this end, we conducted a pilot experiment to find a good set of hyperparameters for the ANN learning algorithm. Hyperparameters of an ANN are variables that are set before optimizing the weights (e.g., number of hidden layers, number of neurons in each layer, learning rate etc.). Our goal was to identify hyperparameters that make predictions matching human behavior on the posttest. Hence, we matched the algorithm's predictions to summary statistics of human performance on the posttest.

Our pilot experiment included 47 undergraduate chemistry students. They were randomly assigned to one of two conditions that used a random training sequence: supervised training ($n = 35$) or unsupervised training ($n = 12$). Participants in the supervised training condition received feedback after each training instance, whereas participants in the unsupervised condition did not receive feedback. We included the unsupervised training condition to generate an evaluation set (used to determine the success of pretraining). This evaluation set was used to pretrain the ANN learning algorithm.

Let there be n supervised human participants. Each participant received a random pretest set, a random training sequence, and a random posttest set. We trained the ANN learning algorithm n times independently (once for each participant). While training for the i-th time we used the training sequence viewed by the i-th supervised human participant. The same posttest set viewed by this participant was also used to evaluate the performance of the ANN learning algorithm after training. Let the error on this posttest set for the i-th human participant and trained ANN learning algorithm be pp_i and pn_i respectively. Then, Equation (3.3) is a measure used to determine whether or not an ANN learning algorithm's performance is comparable to the average human. Note that lower *error rates* are desirable.

$$error\ rates = \left|\frac{1}{n}\left(\sum_{i=1}^{n} pp_i - \sum_{i=1}^{n} pn_i\right)\right| \tag{3.3}$$

Table 3.1 reports the accuracies of participants in the pilot experiment.

Table 3.1 Accuracy in Pilot Experiment by Training Condition. Average pretest, training and posttest accuracy with SEM in parentheses.

Condition	**Pretest**	**Training**	**Posttest**
Supervised	79.9 (1.8)	75.7 (1.2)	89.4 (1.4)
Unsupervised	77.9 (3.4)	78.5 (2.8)	77.1 (3.3)

We note that, on the one hand, humans usually have some degree of prior knowledge about chemistry. On the other hand, the weights of an ANN are generally initialized at random. To model the effect of prior knowledge, we introduced a pretraining phase for the ANN learning algorithm. To this end, we drew a large sample of instances (10000) from the combined test and training distribution ($\frac{1}{2}P_e + \frac{1}{2}P_t$) to form a pretraining set. Further, we combined the pretest instances across both the supervised and unsupervised conditions, along with the training instances in the unsupervised condition to form the pretraining evaluation set. Because we did not provide feedback for these instances, we assumed that the participants did not learn anything new while going through them. Formally, let participants' error on the pretraining evaluation set be called human pretraining error. We then trained the ANN learning algorithm on the pretraining set. Note that an ANN can train over the same set through multiple iterations (formally known as epochs). We trained the ANN learning algorithm until its error on the pretraining evaluation set was smaller than human pretraining error. This concluded the pretraining phase.

Table 3.2 Hyper-parameters for the ANN learning algorithm

Parameter name	**Values explored**	**Best value**
Embedding size	$1,2,4,8,16$	2
Learning rate	$0.00001,0.00005,0.0001,0.0005,0.001,0.005,0.01,0.05,0.1$	0.0001
History window size	$0,1,2,4,8,16,32,60$	2
Backprop count	$1,2,4,8,16$	2
Number of hidden layers before embedding layer	$0,1,2,3,4$	0
Number of hidden units in each column	$10,20,40,80,160$	N/A

We used standard coordinate descent with random restart to find a good hyperparameter set. Coordinate descent successively minimizes the *error rates* along the coordinate directions (e.g., embedding size, learning rate). At each iteration, the algorithm chooses one particular coordinate direction while fixing the other values. Then, it minimizes in the chosen coordinate direction. Table 3.2 shows the values of the hyperparameters over which we decided to explore along with the best value found. These hyperparameters were used to identify the optimal training sequence.

3.1.4 Finding an Optimal Training Sequence

We used the ANN learning algorithm to generate an optimal training sequence for the perceptual-fluency instances. In Equation 3.1, we defined the optimization problem to solve. We solved this problem by searching over the space of all possible training sequences. Without limiting the size of the training sequence, the search space becomes infinite and infeasible. To mitigate this issue, we set the size of the candidate training sequences to 60. This aligns with prior research on perceptual learning [111]:

$$\mathscr{O} = \underset{S \in \mathscr{C}_t, |S|=60}{\operatorname{argmin}} P_{(x,y)\sim P_e}(\mathscr{A}(S)(x) \neq y) \tag{3.4}$$

We used a modified hill climbing algorithm to find such an optimal training sequence. Hill climb search takes a greedy approach. Procedurally, we started with one particular training sequence. Then, we evaluated neighbors of that particular training sequence to determine whether a better one existed. If so, we moved to that one. This process stopped when no such neighbors were found. This search algorithm is defined with its states and neighborhood definition:

- **States:** Any training sequence $S \in \mathscr{C}_t$ of size 60
- **Initial State:** A training sequence selected by a domain expert.
- **Neighborhood of S:** Any training sequence that differs with S by one instance is a neighbor. For computational efficiency, we restricted ourselves to only inspecting 500 neighbors for a given training sequence. We do so by first selecting an instance S uniformly at random. Then we replace the selected instance with 500 randomly selected instances with the same label (i.e., same y value). This made our search algorithm stochastic.

3.2 Human Study with Amazon MTurk Workers

Our main goal was to evaluate whether the optimal training sequence yields higher learning outcomes. To this end, we conducted a randomized, controlled experiment with humans. Here, we discuss our experimental setup and associated results.

3.2.1 Participants

We recruited 368 participants using Amazon's Mechanical Turk (MTurk) [20]. Among them, 216 were male and 131 were female. The rest did not disclose their gender. Most of the

participants were below the age of 45 (86%) and the greatest number (192) fell in the age group $24-35$. Among the 95.4% who disclosed their knowledge about chemistry, around 45.7% had taken an undergraduate-level chemistry class.

3.2.2 Test Set

Because our goal was to assess transfer of learning from the training sequence to a novel test set, we chose training and test instances from separate distributions. Hence, we randomly divided the 142 molecules that we selected for this experiment into two sets of 71 (training molecules, $\mathscr{X}_t$ and test molecules $\mathscr{X}_e$). One of the sets was used to create the test distribution, whereas the other one was used to create the training distribution. We now describe in more detail how we created the test distribution P_e because our goal was to reduce humans' error rates on the test set. We used the following procedure.

- $x_1 \sim p_1$, where p_1 is a marginal distribution on $\mathscr{X}_e$. p_1 is "importance of molecule x_1 to chemistry education" and was constructed by manually searching a corpus of chemistry education articles for molecule text frequency.
- With probability $1/2$, set $x_2 = x_1$ so that the true answer $y = 1$.
- Otherwise, draw $x_2 \sim p_2(\cdot \mid x_1)$. The conditional distribution p_2 is based on domain experts' opinion that favors confusable x_1, x_2 pairs in an education setting. Also note that, $p_2(x_1|x_1) = 0, \forall x_1$. Taken together,

$$P_e(x_1,x_2) = \frac{1}{2}p_1(x_1)\mathbb{I}_{\{x_1=x_2\}} + \frac{1}{2}p_1(x_1)p_2(x_2 \mid x_1).$$

Both the pretest and posttest instances were sampled from this distribution across all conditions.

3.2.3 Experimental Design

We compared three training conditions:

1. In the *machine teaching sequence* condition, we used the optimal training sequence $\mathscr{O}$ found by the modified hill climb search algorithm. For all $(x_1,x_2) \in \mathscr{O}$ (here $x_1 \in \mathscr{X}_t, x_2 \in \mathscr{X}_t$), the corresponding true answer y was the indicator variable on whether x_1 and x_2 were the same molecule: $y = \mathbb{I}_{\{x_1=x_2\}}$. We presented x_1 and x_2 in Lewis and space-filling representations to the human participants, respectively.

Participants gave their binary judgment $\hat{y} \in \{0,1\}$. We then provided the true answer y as feedback to the participant.

2. In the *expert generated sequence* condition, the training sequence was constructed by a domain expert using perceptual learning principles (using molecules only from $\mathscr{X}_t$). Specifically, an expert on perceptual learning sequences visuals (with a decade of experience) constructed the sequence based on the contrasting cases principle [68, 113], so that consecutive examples emphasized conceptually meaningful visual features, such as the color of spheres that show atom identity or the number of dots that show electrons. For example, if one problem presented visuals that showed different molecules, the next problem might present visuals that showed the same molecules. To create such sequences, we randomly set the length of the subsequence that retained one visual to be 1-4 problems. Then we systematically varied visual features that play a role in chemistry learning, as determined by our prior research with novice students and chemistry experts [113, 112]. The rest of this condition was the same as the machine training sequence condition. This training sequence is identical to the initial state of the modified hill climb search algorithm that we used to generate the machine teaching sequence.

3. In the *random training sequence* condition, each training instance (x_1, x_2) was selected from the training distribution P_t with $y = \mathbb{I}_{\{x_1 = x_2\}}$. The training distribution P_t for this condition was induced in the same manner as the test distribution P_e but on the set of training molecules $\mathscr{X}_t$. The rest of the condition was the same as the previous ones.

3.2.4 Procedure

We hosted the experiment on the Qualtrics survey platform [107] using NEXT [62]. Participants first received a brief description of the study and then completed a sequence of 126 judgment problems (yes or no). The instances were divided into three phases as follows. First, participants received a pretest that included 20 test instances without feedback. Second, participant received the training, which included 60 training instances sequenced in correspondence to their experimental condition. During this phase, correctness feedback was provided for submitted answers. Third, participants received a posttest that included 40 test instances without feedback. In addition, one *guard instance* was inserted after every 19 instances throughout all three phases. A guard question either showed two identical molecules depicted by the same representation or two highly dissimilar molecules depicted by Lewis structures. We used these guard questions to filter out participates who clicked

through the instances haphazardly. In our main analyses, we disregarded the guard instances. So that no visual representation was privileged, we randomized their positions (left vs. right).

3.2.5 Results

Of the 368 participants, we excluded 43 participants who failed any of the guard questions. The final sample size was $N = 325$. The final number of participants in the conditions random training sequence, expert generated sequence, and machine teaching sequence were 108, 117 and 100 respectively. Table 3.3 reports accuracy on the pretest, training set, and posttest. See Figure 3.5 for a graphical depiction of the same data.

Table 3.3 Accuracy by Training Condition. Average pretest, training and posttest accuracy with SEM in parentheses.

Condition	Pretest	Training	Posttest
Machine	69.5 (1.1)	63.9 (1.1)	74.7 (1.1)
Expert	71.3 (1.3)	72.4 (1.0)	71.7 (1.0)
Random	69.4 (1.1)	70.3 (1.1)	71.1 (1.1)

Effects of condition on training accuracy First, we tested whether training condition affected participants' accuracy during training. To this end, we used an ANCOVA with condition as the independent factor and training accuracy as the dependent variable. Because pretest accuracy was a significant predictor of training accuracy, we included pretest accuracy as the covariate. Results showed a significant main effect of condition on training accuracy, $F(2,321) = 18.8,\ p < .001,\ \eta^2 = .082$. Tukey post-hoc comparisons revealed that (a) the machine training sequence condition had significantly lower training accuracy than the expert generated sequence condition ($p < .001,\ d = -0.32$), (b) the machine training sequence condition had significantly lower training accuracy than the random training sequence condition ($p < .001,\ d = -0.26$), and (c) no significant differences existed between the human and random training sequence conditions ($p = .592,\ d = 0.05$). In other words, during the training phase, the human and random training sequences were equally effective in terms of accuracy, but the machine training sequence was less effective.

Effects of condition on posttest accuracy Next, we tested whether training condition affected participants' posttest accuracy. To this end, we conducted an ANCOVA with condition as the independent factor and posttest accuracy as the dependent variable. Because pretest accuracy was a significant predictor of posttest accuracy, we included pretest accuracy as a covariate. Results showed a significant main effect of condition on posttest accuracy, $F(2,321) = 5.02,\ p < .01,\ \eta^2 = .023$. Tukey post-hoc comparisons revealed that (a) the machine training sequence condition had significantly higher posttest accuracy than the

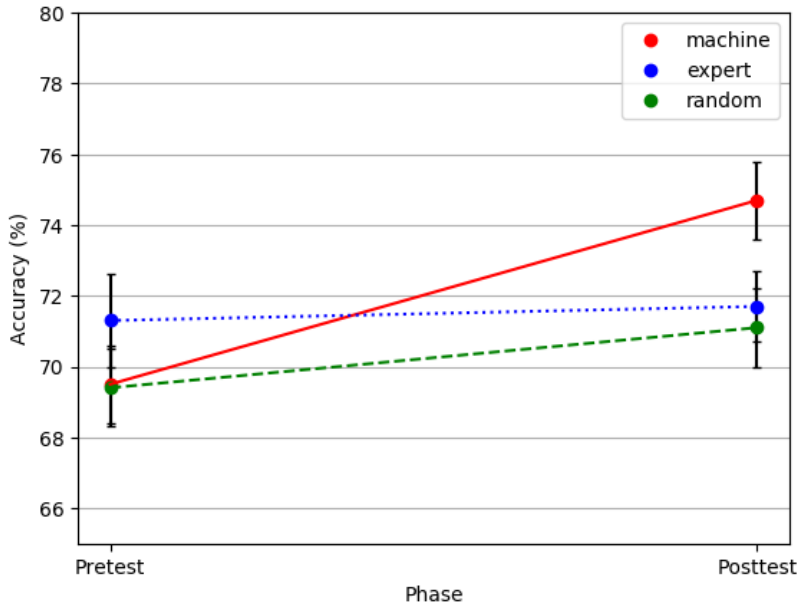


Fig. 3.5 Learning progress between pretest and posttest phases for Amazon MTurk human experiment. The results show an advantage for the machine-generated sequence.

expert generated sequence condition ($p < .05,\ d = 0.16$), (b) the machine training sequence condition had significantly higher posttest accuracy than the random sequence condition ($p < .05,\ d = 0.14$), and (c) no significant differences existed between the human and random training sequence conditions ($p = .960,\ d = -0.02$). In other words, the human and random training sequences were equally effective and the machine training sequence was most effective.

3.3 Human Study with Chemistry Undergrads

We also wanted to see if the results found on Amazon MTurk workers held true for undergraduate chemistry students or not. We conducted another randomized, controlled experiment with undergraduate chemistry students for this purpose. We now discuss out experimental setup and associated results.

3.3.1 Participants

We conducted the study in a 300-level introductory chemistry course for undergraduates. The course is open to freshmen and has a prerequisite of students having completed at least one 100-level chemistry course. However, many students enroll as seniors and have not

taken chemistry since their freshmen year. This indicates that the students in this course have highly variable prior knowledge. 40 undergraduate chemistry students participated in this study.

3.3.2 Procedure

The students received the perceptual-fluency training as a homework assignment with Chem Tutor (Figure 3.6), an intelligent tutoring system (ITS) that provides complex problem with individualized step-by-step guidance [113, 152].

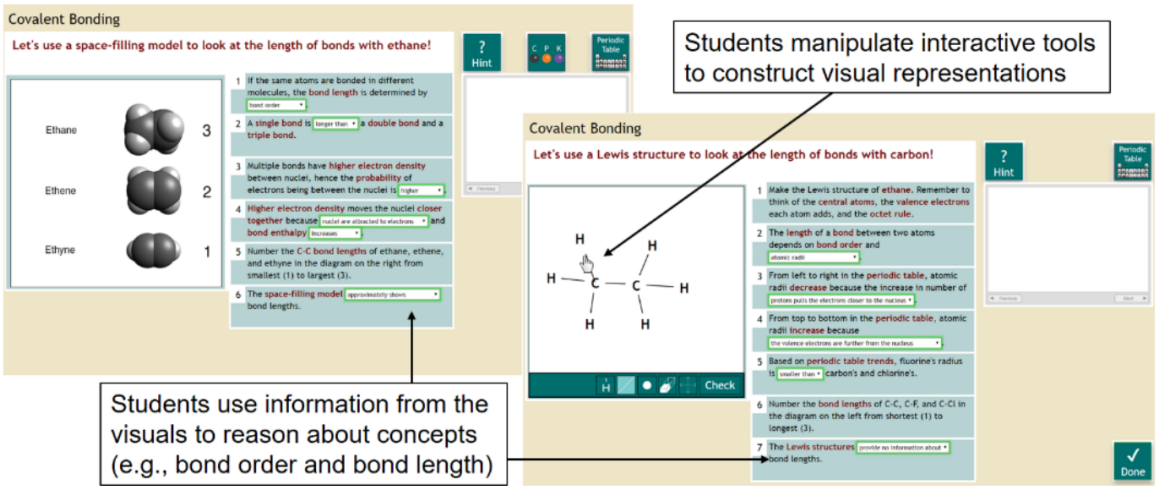


Fig. 3.6 Sample perceptual-fluency problems on Chem Tutor.

The perceptual-fluency training of the assignment was structured as follows. Students first watched a three minute video explaining that they would receive a large number of single-step problems in a row. Students were instructed not to overthink their answer but to intuitively decide if the two visuals showed the same molecule or not. Just like the MTurk human experiment, the instances were divided into three phases: pretest (20 instances without feedback), training (60 instances with feedback) and posttest (40 instances without feedback). However, unlike the previously no guard questions were shown. Students were randomly assigned to one of two training conditions: machine teaching sequence or expert generated sequence. To assess the students' gains in perceptual fluency, we used the same pretest and posttest as in the prior MTurk study. An example of instances displayed on Chem Tutor is presented in Figure 3.7

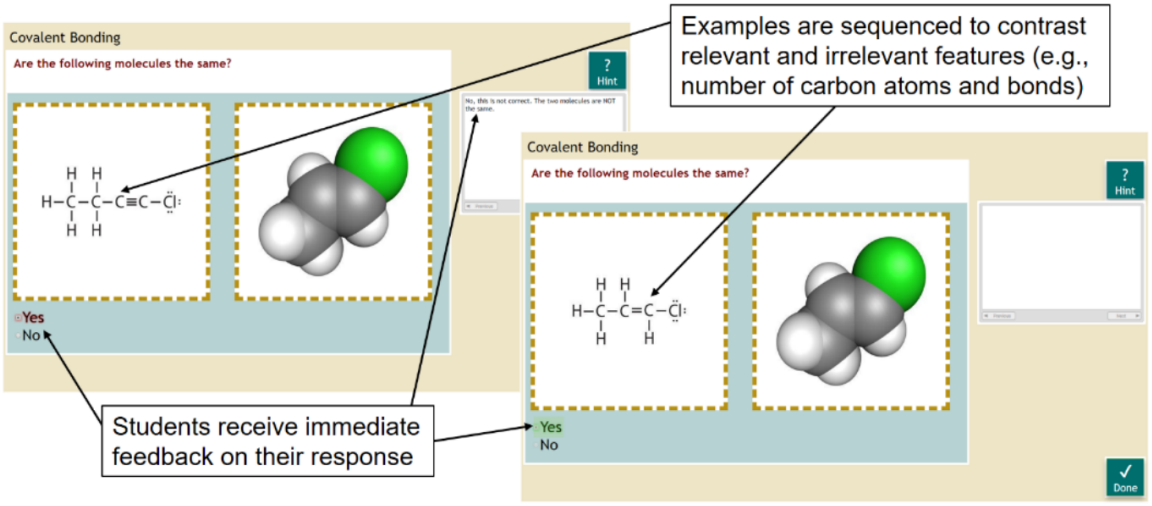


Fig. 3.7 Sample perceptual-fluency problems on Chem Tutor.

3.3.3 Results

Out of the 40 undergrad students, we excluded two as they were statistical outliers on a pretest or posttest yielding $N = 38$ students. Following prior work on efficiency measures [150], we computed perceptual fluency scores as:

$$\text{pereptual-fluency score} = \frac{\text{Z(average correct responses)} - \text{Z(average time per problem)}}{\sqrt{2}} \tag{3.5}$$

Further, to test if the effect of sequence depends on students' prior knowledge, we used the logs from the four interactive instruction activities that students completed prior to the perceptual-fluency problems. We computed prior-knowledge scores as the number of steps students answered correctly on the first attempt. Because the instruction activities ask students to answer questions about chemistry concepts based on the visuals, this measure asses students' knowledge about how the visual shows concepts. We treated prior knowledge as a continuous variable in all analyses.

In the following analyses, we report $p.\eta^2$ effect sizes. We consider $p.\eta^2$ of 0.01 to be a small effect, 0.06 a medium and 0.14 a large effect (following cite). We checked for learning gains using repeated measures ANOVAs with pretest and posttest as dependent measures. Results showed large significant gains in perceptual fluency from the pretest to posttest, $F(1,36) = 8.762, p = 0.005, p.\eta^2 = 0.196$. A multivariate ANOVA showed no significant differences between conditions on the perceptual-fluency pretest or prior knowledge on Chem

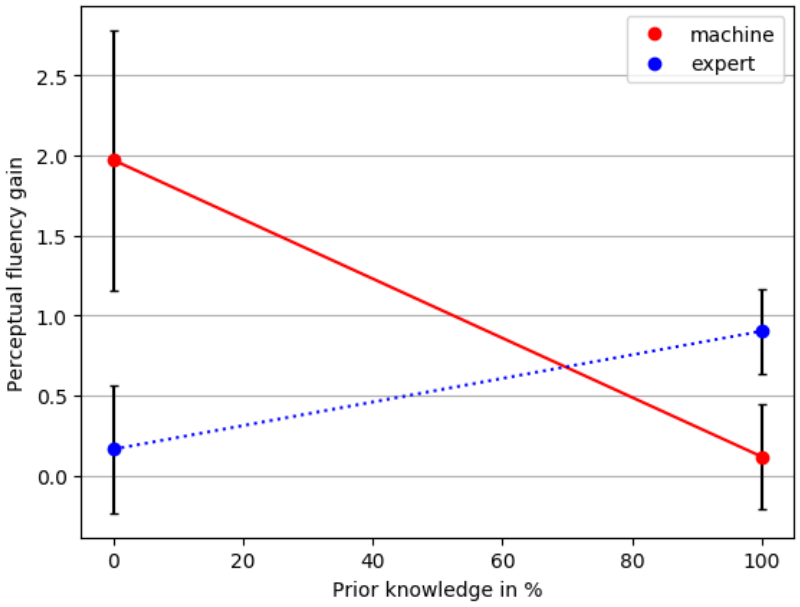


Fig. 3.8 Effect of machine teaching sequence vs expert generated sequence for undergraduate students. The y-axis shows pre-post gains in perceptual fluency scores based on the efficiency measure in 3.5. The x-axis shows prior knowledge. Error bars show standard errors of the mean.

Tutor's interactive instruction activities (Fs < 1). Furthermore, there were no differences between conditions in terms of how much time students spent on the perceptual-fluency training ($F < 1$).

To test if the machine-learned sequence yielded higher gains in perceptual fluency than an expert-generated sequence, we used an ANCOVA model with condition as independent factor, perceptual-fluency pretest and prior knowledge as covariates, and perceptual-fluency posttest as dependent measure. To test if the effects depend on students' prior knowledge, we added an interaction between condition and prior knowledge to the model. In line with prior knowledge on aptitude-treatment interactions [16], we did not dichotomize prior knowledge but modeled the interaction between condition and the continuous prior-knowledge variable.

Results showed a medium-sized significant main effect of condition, $F(1,33) = 4.699$, $p = 0.037$, $p.\eta^2 = 0.125$, such that the machine teaching sequence yielded higher gains in perceptual fluency than the expert generated sequence. The main effect was qualified by a medium-sized significant interaction of condition with prior knowledge, $F(1,33) = 4.788, p = 0.036, p.\eta^2 = 0.127$. The machine teaching sequence was more effective for students with lower prior knowledge while the expert generated sequence was more effective for students with higher prior knowledge (Figure 3.8).

3.4 Discussion

Our goal was to investigate whether a novel educational data mining approach can help identify a training sequence of visual representations that enhances students' learning from perceptual-fluency instances. To this end, we applied the machine teaching paradigm. It involved gathering data from human students learning from perceptual-fluency instances. Next, we generated a cognitive model that mimics human perceptual learning. We then used the cognitive model to reverse-engineer an optimal training sequence for a machine-learning algorithm. Finally, we conducted two experiments: one that compared the machine teaching sequence to a random sequence and to a principled sequence generated by a human expert on perceptual learning for Amazon MTurk Workers, and another that compared the machine teaching sequence to the expert generated sequence for undergrad chemistry students. Results on Amazon Mturk workers showed that the machine teaching sequence resulted in lower performance during training, but greater performance on a posttest.Results on the undergrad students showed that the machine teaching sequence can help weaker students with lower prior knowledge.

These findings make several important contributions to the perceptual learning literature. First, our results can inform the instructional design of perceptual-learning instances. Even though prior research yields principles for effective sequences of visual representations, numerous potential sequences can satisfy these principles. Our results show that this new educational data mining approach can help address this problem. Given a learning algorithm that constitutes a cognitive model of students learning a task, instructors can identify a sequence of instances that likely yields higher learning outcomes.

Our research also show that machine teaching can enhance the effectiveness of perceptual fluency trainings for low-performing students in a realistic educational context. The results on undergrad chemistry students also show that the machine-learned sequence yield higher gains in perceptual fluency than an expert generated sequence for students with lower prior knowledge. This finding shows that our machine learning approach is an effective method for developing perceptual fluency trainings that are attuned to the needs of students whose needs may not be obvious to instructional designers. Our findings also suggest that students with low prior knowledge require different types of training than those with high prior knowledge. Our qualitative comparison between the machine teaching sequence and the expert generated sequence suggests that students with low prior knowledge may benefit from sequences that draw attention to visual features that may seem obvious to experts, such as the mapping between letters and colors. Given that the students in our experiment likely had some exposure to the visuals in prior chemistry courses, we think it is unlikely that they did not know that these features are important. Rather, they may not have been efficient at perceiving

these features. Furthermore, the machine teaching sequence did not repeat visual across consecutive problems, whereas the human generated sequence did. Such repetitions assume that students recall the visuals from previous problems, which is cognitively demanding. Hence, students with low prior knowledge may benefit more from the reduce of cognitive load. Finally we found that the human teaching sequence is more effective for students with high prior knowledge. This finding replicates prior research on the effectiveness of human generated sequence for advanced students. A new contribution of our findings is that we found that students' performance on prior instructional activities with visuals predicts if they have the prerequisite knowledge to benefit from an expert generated sequence or if they should receive a sequence that was machine-learned based on data from novice students to prevent expert blind spot biases.

Our findings also contribute to the educational data mining literature. We provide the first empirical evidence that a ANN learning algorithm constitutes an adequate cognitive model of learning with visual representations. As far as we know, the machine teaching paradigm has thus far only been applied to learning with artificial visual stimuli that vary on only one or two dimensions (e.g. Gabor patches [46]). Thus, our study provides the first demonstration that machine learning along with machine teaching is a viable approach to modeling and improving learning with realistic, high-dimensional visual representations like Lewis structures and space-filling models of chemical molecules. Therefore, we believe this approach is valuable for educational data mining research.

3.5 Limitations and Future Directions

Our findings should be interpreted against the background of the following limitations. First, the search algorithm we used to find the machine teaching sequence did not test all possible training sequences of size 60. As mentioned previously, we only inspected 500 neighbors (out of a potential $5040 = 71 \times 71 - 1$) for any given training sequence. Moreover, we stopped the search algorithm after a predetermined amount of time. We chose this inexhaustive approach because exhaustively finding a solution is not computationally feasible. Thus, we settled for a suboptimal training sequence that still yielded a small risk on the test distribution. Consequently, it is possible to find a better training sequence than the one we used in our experiments.

Second, while determining the hyperparameters of the ANN learning algorithm such that it mimics human perceptual learning, we only searched over a subset of all possible hyperparameters. As a result, it is possible that a better set of hyperparameters exists. Our study was also limited in that we did not account for individual prior knowledge. Hence,

future research needs to investigate how to expand the approach presented in this paper to modeling individual prior knowledge (e.g., for adaptive teaching or personal training).

A third limitation of the present experiments is that our study was constrained in the use of chemistry representations as stimuli. While we used realistic representations that are more high-dimensional than prior perceptual learning studies [46, 36, 148], the complexity of the representations we considered does not reflect all realistic stimuli. Sparser and richer visuals exist and it is possible that machine teaching may yield greater benefits for sparser visuals. We will investigate this hypothesis in future studies.

Finally, we did not contrast the characteristics of machine teaching and expert generated sequences that may account for our results. For example, we did not test if the repetition of visual representations across problems is effective for students with high vs low prior knowledge. Yet, our findings provide first indications that these characteristics may affect the acquisition of perceptual fluency, which can be systematically tested in future research. Future research should also test whether gains in perceptual fluency for low-performing students translates into an enhanced ability to use the visual representations to learn content knowledge.

3.6 Contribution

In this project, I helped in formulating the optimization problem and the hill climb based solution. I also executed the optimization and helped in the human data collection from Amazon Mechanical Turk. Finally, I helped in analyzing the results. The theoretical motivation related to visual representations presented in this work was provided by Martina Rau, Educational Psychology, University of Wisconsin-Madison. The work presented in this chapter has been published at "Ayon Sen, Purav Patel, Martina A. Rau, Blake Mason, Robert Nowak, Timothy T. Rogers, and Xiaojin Zhu. ‘Machine Beats Human at Sequencing Visuals for Perceptual-Fluency Practice.’ International Educational Data Mining Society (2018)" and "Martina A. Rau, Ayon Sen, and Xiaojin Zhu. ‘Using Machine Learning to Overcome the Expert Blind Spot for Perceptual Fluency Trainings.’ International Conference on Artificial Intelligence in Education (2019)".

Chapter 4

Learning to Read

4.1 Introduction

Experiences unfold through time, and learning happens along the way. How learning events are sequenced has an important impact on knowledge acquisition. Educational curricula incorporate sequential structure at multiple scales such as the organization of a single class, the arc of a semester, or the trajectory of a degree-granting program. Our research addresses whether machine learning outcomes can be improved by optimizing the sequence of learning experiences in a complex knowledge domain. Gains in the efficiency of learning could mitigate limitations on human learning arising from perception, attention, memory, and other aspects of human cognition [12, 115].

The production of serially ordered behavior has been studied since Lashley's [81] classic work (see, e.g., [15]). Interleaving is a simple example of how ordering of learning experiences affects learning outcomes [43, 73, 78, 96, 110, 123]. In other cases, the sequence is determined by structure of the to-be-learned material. In elementary mathematics, for example, relations among addition, multiplication, and division dictate the order in which they are taught, allowing instruction to emphasize how one concept or operation participates in understanding the next, more sophisticated one. Other work has suggested that starting simple and increasing problem complexity over time can help people learn more quickly while emphasizing structure that supports generalization [12]. Benefits have been demonstrated in a variety of learning paradigms, including computational experiments involving shape recognition and other perceptual tasks, as well as language [39, 101, 122].

Our research examined whether the sequence of learning trials could be optimized in an artificial neural network trained on a complex problem: learning the correspondences between the written and spoken forms of words in English. Mastering these correspondences is an important step in becoming a skilled reader. The material is difficult to teach and there is

little agreement about how to do it. Many children struggle at this early stage, with negative downstream effects on literacy and life experiences [129]. Spelling-sound correspondences in English are systematic (letters and letter combinations represent sounds) but inconsistent (numerous forms deviate from central tendencies). This quasiregular structure [130] exists at most levels of language [21]. The spelling-sound system does not exhibit any obvious internal structure on which to base a learning sequence. The system consists of numerous patterns differing in unit size, frequency, and consistency across words. The question then is whether the sequence of learning experiences (training trials) can be ordered in a manner that facilitates acquiring this foundational knowledge.

In this paper, we approach sequence optimization as a gradient based optimal control problem for machine learning (i.e., machine teaching [174]). Given a learning algorithm, a pool of examples to choose from, and a target model to train, a machine teacher seeks to design a curriculum that conveys the solution efficiently [171]. However, most of the problems tackled so far in this domain work under a batch setting [171, 84] or for short training sequences [131].

Our objective is to discover a training sequence such that the generalization performance of the learner is maximized given a fixed sequence length of 10,000 words. Critically, this is too short to establish reading proficiency—a critical constraint analogous to the time pressure experienced by children. Yet, the sequence is long enough to represent a useful amount of experience in early development and to pose a hard combinatorial problem, given that we explore sequences with as many as 1,000 unique words in any order. A sequence of this length represents the quantity of words a child would experience from reading ≈ 80 books appropriate to an early reader.

The main contributions of this paper are as follows. First, we formalize the problem of learning to read aloud as an optimal control problem. We then show how a teacher can solve this problem in two steps: 1) using stochastic gradient descent to find optimal distribution of the words at different steps (*time varying distribution*) of the sequence, 2) sampling an optimal sequence from the optimal time varying distribution. Finally, we experiment with two ecologically valid vocabularies composed of either words a child or an adult is likely to encounter. Our results show significant improvement over training sequences sampled using metrics that reflect the prevalence of words in a child or adult's language environment.

4.2 Preliminaries

4.2.1 Vocabulary and Feature Vectors

We represent a vocabulary of monosyllabic words with V. Each word in V has an orthographic input ($\mathbf{o} \in \{0,1\}^{260}$) and a phonological output ($\mathbf{y} \in \{0,1\}^{200}$) representation. The input and output patterns can represent up to 10 letters and 8 phonemes, respectively. Each letter is encoded as a 26 dimensional one-hot vector. Each phoneme is encoded as a 25 dimensional vector of articulatory features that accommodate all English speech sounds. Each phoneme is encoded as $\mathbf{m} \in \{0,1\}^{25}$, such that each articulatory feature is either set or not.

Words are encoded such that the first vowel always occurs in the fourth letter or phoneme position. For example $V_{\text{coals}} = (\texttt{\_\_coals\_\_\_}, \texttt{\_\_kolz\_\_})$. Notice that `oa` maps to the single vowel phoneme `o`. The fifth letter position was also reserved for vowel representation: $V_{\text{duct}} = (\texttt{\_\_ca\_t\_\_\_\_}, \texttt{\_\_k@t\_\_})$. Note that padding '_' is denoted by zero vectors in both input and output.

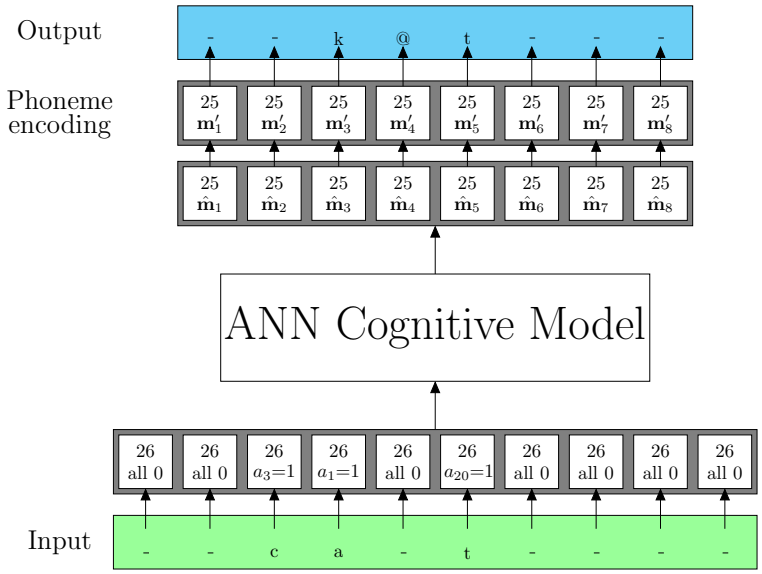


Fig. 4.1 ANN cognitive model. It takes as input the orthographic representation of a word and predicts the phonological representation. a_i indicates the i-th position of the input character vector. The continuous output vector is first decoded into individual phonemes, and then the complete phonological representation.

4.2.2 Cognitive Model

We employ an ANN architecture with a long history of relevance in the cognitive science literature on reading development (Figure 4.1) [106, 130]. It is a fully-connected feed-forward network with a single hidden layer (100 units) and sigmoid activation functions on all hidden and output units. Our research builds on previous efforts to study teaching within a connectionist framework [56], specifically examining whether the sequence of learning trials can be optimized to support development. The learner's environment consists of monosyllabic words that are presented one at a time, all letters simultaneously in parallel. This allows for the use of a relatively simple, non-recurrent network while preserving essential aspects of early visual word recognition and its development, namely that a monosyllabic word can be taken in on a single visual fixation.

The learning procedure is defined by the dynamics $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$. Here, $\mathbf{x}_t \in \mathscr{X}_t$ is the state of the model before training round t. In our setup, it is the vector of weights of the ANN. The control input $\mathbf{u}_t$ is the orthographic, phonological representation pair for a word in the training pool: $\mathbf{u}_t \equiv (\mathbf{o}_t, \mathbf{y}_t) \in U \subset V$. This item is picked by a teacher and presented to the learner for training at round t. The function f defines the evolution of the state under external control. Here, it is a backpropagation function with Nesterov momentum and cross entropy loss function. The training round t ranges from 0 to $T-1$ where the time horizon T is fixed at 10K.

After training is complete, the model can be used to make predictions denoted by $\hat{\mathbf{y}} = \mathscr{A}(\mathbf{x}_T, \mathbf{o})$. Here $\hat{\mathbf{y}} \in [0,1]^{200}$. We decode $\hat{\mathbf{y}}$ by first identifying individual phonemes. Let M denote the set of all possible phonemes (including padding). Given a continuous vector $\hat{\mathbf{m}}_j \in [0,1]^{25}$, we decode it to a phoneme by $\mathbf{m}'_j = \text{argmin}_{\mathbf{m} \in M}(\|\hat{\mathbf{m}}_j - \mathbf{m}\|_2)$. Let $\mathbf{y}' \equiv [\mathbf{m}'_1, \ldots, \mathbf{m}'_8]$ i.e., $\mathbf{y}'$ denote the concatenation of the decoded phonemes. Then the prediction is correct if $\mathbf{y}' = \mathbf{y}$. We use the function $\rho : [0,1]^{200} \to \{0,1\}^{200}$ to denote the complete decoding procedure i.e., $\mathbf{y}' = \rho(\hat{\mathbf{y}})$.

4.2.3 Teacher's Cost Functions

The teacher takes into consideration two separate costs while designing a training sequence: a running cost and a terminal cost. The running cost (denoted by $g_t(\mathbf{x}_t, \mathbf{u}_t)$) identifies how difficult/easy a problem is to teach to the learner. The terminal cost on the other hand is defined on the final state of the learner i.e., the trained model:

$$g_T(\mathbf{x}_T) = \frac{1}{|E|} \sum_{(\mathbf{o},\mathbf{y}) \in E} \mathbb{1}(\rho(\mathscr{A}(\mathbf{x}_T, \mathbf{o})) \neq \mathbf{y}) \tag{4.1}$$

Here, $E = V - U$ is a test set and $\mathbb{1}(\cdot)$ is the indicator function. The terminal cost estimates how well the learner generalizes to unseen examples.

4.3 Optimal Control Problem

In this section we define the teacher's optimal control problem and propose a solution. The teacher's objective is to find a sequence of control inputs (training example sequence) that reduces the total cost. We consider all examples to be equally difficult/easy. As the time horizon T is fixed, total running cost is also fixed. This reduces the teacher's objective to

$$\begin{aligned} \min_{\mathbf{u}_0,\ldots,\mathbf{u}_{T-1}} \quad & g_T(\mathbf{x}_T) \\ s.t. \quad & \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), \mathbf{u}_t \in U, \forall t \\ & \mathbf{x}_0 \text{ given} \end{aligned} \tag{4.2}$$

We propose a gradient based solution to this problem in two steps. We start by defining a *time varying distribution* over the examples in U. We assume the teacher has a *start multinomial* $P = (p_1, \ldots, p_K)$ and an *end multinomial* $Q = (q_1, \ldots, q_K)$ over U. Here, $K = |U|$. At training round $t = 0, \ldots, T$, the teacher uses an interpolated multinomial:

$$R_t = (T - t)/T * P + t/T * Q \tag{4.3}$$

The teacher will then draw $\mathbf{u}_t \sim R_t$ and train the learner with this example. (P, Q) then defines the time varying distribution. We call this a time varying distribution as R_t changes at each training round. We find an optimal value for this pair: $(P^*, Q^*) = \text{argmin}_{(P,Q)} E[\psi(P, Q)]$. Here $\psi(P, Q)$ denotes the terminal cost of a sequence drawn from the interpolated multinomials in (4.3) defined by (P, Q). As $\psi(P, Q)$ is stochastic, we minimize over the expected terminal cost. After identifying (P^*, Q^*) we sample multiple sequences from the time varying distribution and pick the best one to solve (4.2).

This two step procedure has multiple advantages over solving (4.2) directly. First, (4.2) is a combinatorial optimization problem. Such problems are hard to solve in practice, especially for long sequences. Using a time varying distribution on the other hand allows us to optimize over a continuous space. Moreover, directly solving the combinatorial optimization problem does not allow us to easily identify why a particular sequence is better than others. But finding (P^*, Q^*) allows us to readily understand different properties of good sequences. For example, by inspecting P^*, Q^* and R_t^* we can identify examples that are more important during the various training rounds.

Because (P, Q) are themselves multinomial distributions, they have a natural normalization constraint and are bounded. We circumvent this issue by reparametrizing P as $(\alpha_1, \ldots, \alpha_{K-1}, \alpha_K = 1) \in \mathbb{R}^K$, unconstrained. Note that the last element is a constant 1. We recover P by $P_i = \exp \alpha_i / \sum_{j=1}^{K} \exp \alpha_i$. Similarly Q is reparametrized as $(\beta_1, \ldots, \beta_{K-1}, \beta_K = 1) \in \mathbb{R}^K$. We now optimize $\mathbf{z} = (\alpha_1, \ldots, \alpha_{K-1}, \beta_1, \ldots, \beta_{K-1}) \in \mathbb{R}^{2K-2}$: $\mathbf{z}^* = \text{argmin}_{\mathbf{z} \in \mathbb{R}^{2K-2}} \mathbb{E}[\ell(\mathbf{z})]$. $\mathbb{E}[\ell(\mathbf{z})]$ denotes the expected value of the terminal cost function when training sequences are sampled using $\mathbf{z}$. We find $\mathbf{z}^*$ by using stochastic gradient descent (sgd) with momentum

$$\begin{aligned} \Gamma_{s+1} &= \gamma \Gamma_s + \eta \nabla \mathbb{E}[\ell(\mathbf{z}_s)] \\ \mathbf{z}_{s+1} &= \mathbf{z}_s - \Gamma_{s+1} \end{aligned} \tag{4.4}$$

Here, η and γ are step size and momentum respectively. Γ_0 is set to 0. $\nabla E[\ell(\mathbf{z}_s)]$ is the gradient of $E[\ell(\mathbf{z}_s)]$ with respect to $\mathbf{z}_s$. We estimate $\nabla E[\ell(\mathbf{z}_s)]$ by using finite difference stochastic approximation[42]:

$$\nabla \mathbb{E}[\ell(\mathbf{z}_s)] \approx \mathbb{E}_{\mathbf{v}}[(\mathbb{E}[\ell(\mathbf{z}_s + \delta \mathbf{v})] - \mathbb{E}[\ell(\mathbf{z}_s)])\mathbf{v}](2K-2)/\delta \tag{4.5}$$

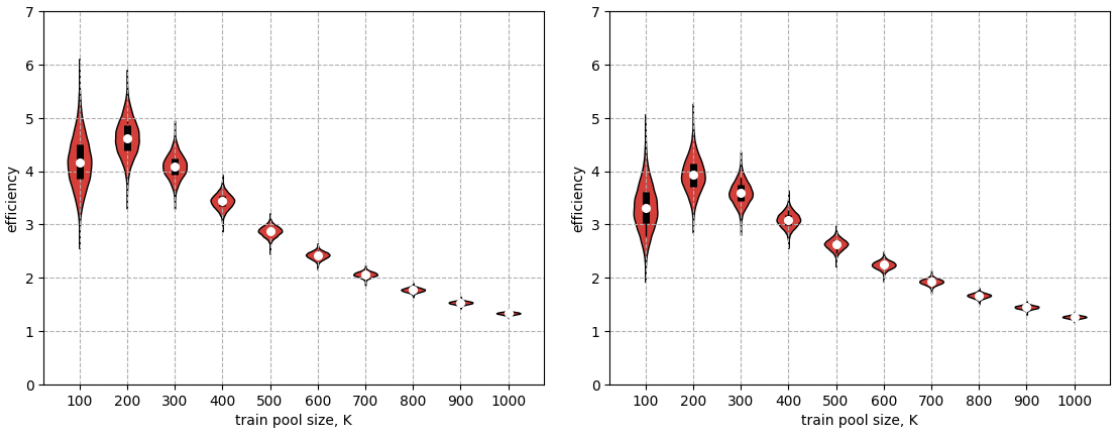
Here, $\mathbf{v} \in \mathbb{R}^{2K-2}$ is a uniformly random unit vector and $\delta > 0$. Note that the outer expectation is taken over $\mathbf{v}$.

4.4 Experiments and Results

In this section, we empirically evaluate our proposed solution. We compared our results against multiple baselines where the words are sampled with respect to their natural frequency in the environment. We ran sgd with momentum in two steps. First we found an optimal distribution ($\bar{P}^*$) from which the words can be drawn. Note that this is not a time varying distribution. Here, $\bar{P}_0$ was initialized as a uniform vector. We ran sgd with momentum again to find the optimal time varying distribution by initializing (P_0, Q_0) with $(\bar{P}^*, \bar{P}^*)$.

4.4.1 Datasets

We constructed two different training corpora of monosyllabic words, one using prevalence statistics relevant for adults and the other for children. Child words were selected if they appeared at least twice in the corpus of 250 children's books, and if a word possessed an age of acquisition (AOA) rating of 9 years old or younger from relevant norms [79]. Additionally, words needed to have a string length of greater than one and contain an orthographic vowel (e.g., the word "hmm" was discarded despite being included by the above criteria). This

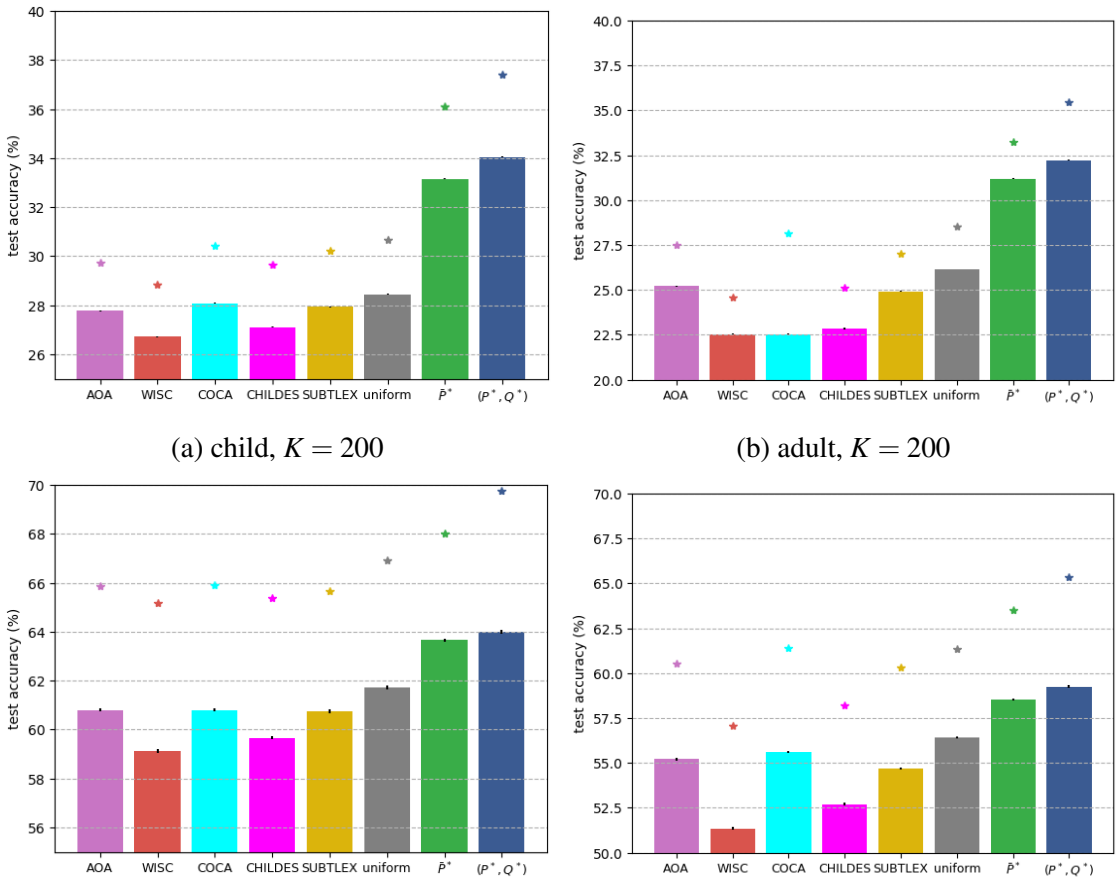


(a) child (b) adult

Fig. 4.2 Efficiency for different training pool sizes. Average efficiency peaks at training pools of size 200. The black bar shows the 25th and 75th percentile.

yielded a vocabulary of 2869 words in total. In order to have a corresponding set of words for comparison to the primary child set, we followed a similar procedure using prevalence statistics for adult texts. The top 1000 most frequent monosyllabic words were selected from the Corpus of Contemporary American English (COCA) [32] regardless of their presence in the child set. An additional 1869 words were selected by their rank frequency past the 1000th most frequent monosyllabic word in COCA, skipping words that were present in the child set. This sampling strategy was used in order to minimize the dependence of the two sets without skipping too many very common words. This resulted in a comparison set with 947 words (34%) also appearing in the child set.

Each vocabulary is divided into a training pool and test set. First, we chose how many words should be a part of the training pool, evaluating training pools of different sizes. For a particular size K, we randomly divided the vocabulary into a training pool and a test set. Then we batch trained the ANN cognitive model on this training pool to convergence and calculated the number of words that are correctly predicted by the trained model in the test set. We represent this value with c. For this batch training we used a learning rate of 0.1 for faster convergence. We evaluated efficiency of size K as c/K. This process was repeated 100K times to find the average efficiency for a particular K. We performed this experiment for $K = \{100, 200, \ldots, 1000\}$. The results are shown in Figure 4.2. It can be seen that on average $K = 200$ is the most efficient and hence this particular K value is used. We also used $K = 1000$ for further experiments as children are expected to encounter a larger variety of words in a classroom setting. For a fixed K, we chose the training pool/test set split that resulted in the largest test set accuracy in the aforementioned experiment.



(a) child, $K = 200$

(b) adult, $K = 200$

(c) child, $K = 1000$

(d) adult, $K = 1000$

Fig. 4.3 Average test accuracy for different distributions sampled over 1000 sequences. The error bars show standard error. The optimal sequence test accuracy for each distribution is presented with an '*' above each bar.

4.4.2 Hyperparameters

For the ANN cognitive model we used a learning rate of 0.02 and a Nesterov momentum of 0.9. These values were motivated by previous explorations in this domain [28]. For sgd with momentum in (4.4), we set $\eta = 0.01$ and $\gamma = 0.9$. We set $\delta = 0.01$ to estimate $\nabla E[\ell(\mathbf{z}_s)]$ in all experiments. We approximated $\mathbb{E}_{\mathbf{v}}[(\mathbb{E}[\ell(\mathbf{z}_s + \delta\mathbf{v})] - \mathbb{E}[\ell(\mathbf{z}_s)])\mathbf{v}]$ in (4.5) by sampling 20 and 100 uniform random unit vectors $\mathbf{v}$ for $K = 200$ and $K = 1000$ respectively.

4.4.3 Baselines

To compare our results we considered multiple baselines drawn from data that convey the prevalence of words in our experimental corpora in speech and print, many of which were

also used for sampling words for inclusion in the corpora. Given that printed and spoken language varies according to the audience for which it is designed, we gathered data for our words relevant to both adult and child audiences and from spoken and printed sources. Child-directed speech data were taken from the CHILDES database [92]. A baseline derived from word frequencies in child-directed text was also used, drawing from the same source that was used in sampling our child-directed words described previously (The Wisconsin Children's Book Corpus). AOA data [79] were used for estimates of the age at which words were learned from non-print based sources. A baseline from adult-directed text frequencies utilized the COCA [32], a common large database used for text frequencies. Paralleling the child-directed sources, values for child-directed speech were also used as a baseline, utilizing data from the SUBTLEXus databases [19]. While SUBTLEXus isn't strictly a speech database, these data represent usage in movies, which we take to be speech-like. Additionally, we implement a baseline using a uniform sampling probability over all candidate words, not favoring any particular word in the selection process. The sampling probabilities drawn from these baselines serve as relevant comparisons for our optimized probabilities.

4.4.4 Results

The expected test set accuracy of the optimal distributions, along with the baseline models, are presented in Figure 4.3. These results show that we were able to find time varying distributions that are significantly better than the baselines. The gains are higher for $K = 200$ than $K = 1000$. It should also be noted that the overall test accuracy values found using $K = 200$ are significantly lower than that of $K = 1000$. This is not surprising given that more irregular words are incorporated into the $K = 1000$ training pool, meaning that they are not in the generalization set. But in all cases the average test accuracy for (P^*, Q^*) is statistically better than that of $\bar{P}^*$ and the other baselines ($p < 0.001$ for student's t-test). The corresponding optimal sequence test set accuracy values are presented with '*' in the same Figure. Not surprisingly, the optimal sequence found using (P^*, Q^*) is always the best one.

In trying to determine those aspects of words that make them well suited to enhance learning, we calculated a number of word-level variables that might help us understand the distributions of P^* and Q^*. Orthographic neighborhoods were calculated as Levenshtein Distance with $D_{\text{Lev}} = 1$. Phonological neighborhoods were determined by the number of words sharing the same orthographic body (i.e., the *ushed* in *hushed*) and phonological rime. For example, *rushed* is a neighbor of *hushed* but *pushed* is not, despite sharing its body. Phonological density refers to the number of features on (i.e., equal to one) for a given word's target phonological representation. Finally, three measures of Shannon Entropy were calculated, representing estimates of the predictability of a given orthographic unit's

associated phonological code[135]. The entropy of the oncleus unit (orthographic onset plus orthographic nucleus) is calculated with respect to any consonants that come before the orthographic vowel segment plus the vowel segment (i.e., the *broo* in *brook*). Vowel entropy is calculated for the orthographic vowel segment (i.e., the *oo* in *brook*). And the rime entropy is calculated for the orthographic vowel and everything that follows it (i.e., *ook* in *brook*). Vowel entropy is calculated for the orthographic vowel segment (i.e., the *oo* in *brook*), and the rime entropy is calculated for the orthographic vowel and everything that follows it (i.e., *ook* in *brook*). Three word prevalence measures are also included (AOA, child text frequency, and adult text frequency).

Table 4.1 Correlations between word-level variables and mean of P^* and Q^*. Correlations calculated as Spearman's ρ, and bolded if $p < .05$.

Variable	Child, $K = 200$	Adult, $K = 200$	Child, $K = 1000$	Adult, $K = 1000$
Orthographic length	0.01	**0.16**	0.02	0.01
Phonological length	0.1	**0.23**	0.05	0.02
Orthographic neighbors	0	-0.1	-0.05	0.02
Phonological neighbors	0.02	-0.08	**-0.11**	-0.03
Phonological density	0.11	**0.15**	0.05	0.02
Morphology	0.05	**-0.14**	0.05	0.02
Oncleus entropy	0.03	**0.14**	-0.01	0.02
Vowel entropy	0.07	**0.22**	0.03	0.02
Rime entropy	**-0.18**	0	-0.05	0
Age of acquisition	0.05	0.11	0	-0.01
Child text frequency	-0.02	-0.12	0.04	0.02
Adult text frequency	0.03	-0.12	-0.04	0

A few observations can be made about these correlational results in an attempt to understand what makes a word beneficial for learning. Orthographic and phonological length correlate with average sampling probability in the adult, $K = 200$ condition. Some conditions tended to optimize for the predictability of subword orthographic units with respect to their phonology, namely the models trained on a candidate pool of 200 words. However, the particular unit differs across conditions. The child, $K = 200$ condition is associated with higher sampling probabilites for rimes (word endings), and the corresponding adult condition with word-initial segments (onclei) and vowels. This connects with findings in the reading development literature that have documented varying and sometimes conflicting findings about the location of statistical regularities that influence behavior in reading development [135, 154]. Little is explained by the structural properties included here for the conditions in with larger candidate pools. Only the number of phonological neighbors seem to be related to mean of P^* and Q^* and only in the child model for the models trained with 1000 eligible words.

4.5 Conclusion

We have demonstrated a gradient based optimal control approach for discovering long sequences that achieves efficiency gains above and beyond batch optimization and frequency-weighted sampling. Compared to the prior state of the art, which was restricted to applications with short sequences, successful optimization over $T = 10,000$ with as many as $1,000$ unique elements is a noteworthy advance.

The learning environments employed in our simulations were constructed for their relevance to development as established from prevalence statistics for child and adult print and speech. We find an optimal time varying distribution defined using two distributions over training words, which can be used to establish good training sequences for the learner. Having done so in a cognitive architecture that simulates visual word recognition and development [130, 106], this is potentially a first step towards practical applications in reading education [129].

While variability in selection probabilities assigned to words across candidate pools indicate words that are useful for learning and performance, the structural properties that lead to their utility need to be studied further. The distributions for P^* and Q^* across conditions aren't substantially different in the optimization attempts reported here, despite success in optimizing relative to several motivated comparison conditions including a P^* only distribution. In future work, more needs to be done to explore the effects of defining the time varying distribution using more than two distributions, positioned in ways throughout the sequence that may show advantages over the linear interpolation scheme employed here. This includes experimentation with non-linear interpolation functions over the distributions to define R_t.

Nonetheless, our results show the possibility and potential of optimizing the sequence of words an early developing reader is exposed to in order to enhance learning, including generalization to untaught words.

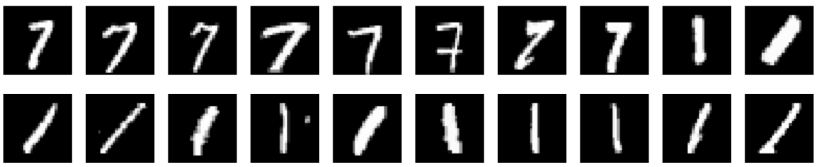
4.6 Contribution

In this project, I helped formulating the optimal control problem. I also took part in determining the two part solution. Moreover, I performed the optimization helped with the analysis of the results. The theoretical motivation related to reading development presented in this work was provided by Christopher Cox, Department of Psychology, Louisiana State University; Matt Cooper-Borkenhagen, Department of Psychology, University of Wisconsin-Madison and Mark Seidenberg, Department of Psychology, University of Wisconsin-Madison.

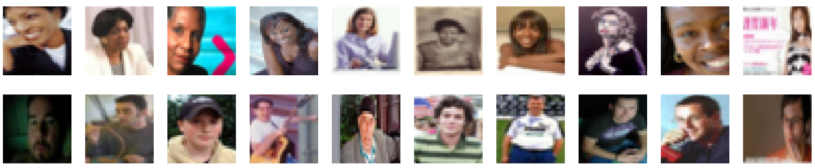
Chapter 5

Training Set Camouflage

Look at the classification training set shown in Figure 5.1a. If you think that the task is handwritten digit classification then you have already been successfully attacked, in a sense to be made precise below. The actual task is to classify woman vs. man (samples shown in Figure 5.1b). A standard logistic regression learner trained on the images in (a) achieves high gender classification accuracy on the images in Figure 5.1b.



(a) Camouflaged training set



(b) Secret classification task

Fig. 5.1 Example of training set camouflage

More broadly, we consider an agent Alice who has a secret classification task (e.g., classifying images of men and women) and a corresponding private training set (men and women images). Alice wants to train a second agent, Bob, on the secret task. However, the communication channel between them has an eavesdropper we refer to as a third agent Eve. Eve expects Alice to send data in some particular format (e.g., image or text). She also does not expect Alice to send anything out of the ordinary. If that is the case then she can stop

the communication between Alice and Bob completely. In this scenario sending the private training set will reveal Alice's intention; sending the model parameters directly will also raise suspicion. Alice must camouflage the communication for it to look mundane to Eve, while avoiding excessive coding tricks with Bob beforehand. In the present work, we show how Alice can construct a camouflaged training set on a *cover task* which (i) does not look suspicious to Eve, and (ii) results in Bob learning an accurate model for the secret task.

Hiding information in plain sight such that its presence is not suspected is known as steganography. Steganography is not new. In the fifth century BCE messengers would have their heads shaved and a message written on their scalp. Regrowing their hair served to hide the message which would only be revealed because the intended recipient knew to shave the messenger's head [75]. In more modern times, steganographic techniques are used to detect unauthorized distribution of digital media [29]. Note that, steganography is different crypotgraphy [67, 151], where the goal is to hide the data content. In cryptography, the communicating agents have access to some particular key (pairs) which is used to encrypt and decrypt data. Cryptography cannot be used if someone monitoring the data can alter or stop the data passing. In such cases, steganography becomes important because we do not want any intervening eavesdropper to become suspicious and may even stop the transmission entirely.

While developing a procedure of hiding information, the role and capabilities of the eavesdropper become key. In a generic steganographic setting, the eavesdropper is known as the attacker which is different from the terminology we use (Alice is the attacker in our scenario). The analysis of data by the eavesdropper Eve to find the existence of hidden data is known as steganalysis. Eve can be either passive and merely observes traffic [22], or active and tries to modify the hidden message [108, 26]. In this manuscript we assume a passive observer. To our knowledge, steganography for machine learning is new. We note that training set camouflage differs from so called "poisoning attacks" [71] in two primary ways: (i) Alice aims to communicate information to Bob about a potentially completely unrelated task, not affect his behavior on the original task and (ii) Alice specifically aims to avoid detection by Eve.

Due to the widespread use of machine learning in sensitive fields ranging from social media to health care, the study of the security ramifications of using ML techniques is well studied [10, 88]. The work presented herein adds to this conversation, as we reveal an additional avenue of attack (hence we call Alice the attacker and not Eve). For example, Bob might be a model that classifies job applicants as "should hire" and "shouldn't hire". The company may have many records (collected over years) of job applicants and how they performed. It is expected from Alice to select a subset of these records and present to Bob,

with the idea that training on the complete set is too time consuming. But Alice is a malicious agent and wants Bob to actually learn some additional bias (e.g., racial, gender etc.). Another example would be training an underlying bias to humans. Eve can be viewed as a human moderator agent in a social network setting. The images sent by Alice may seem benign to her but it might be used to make Bob more biased. In such scenarios, Alice will select a subset of records that satisfies her goals while Eve's responsibility is to verify the data sent by Alice to Bob. Our specific contributions in this paper are as follows: (i) We propose a general mathematical framework for defining how Alice can achieve training set camouflage. (ii) We empirically show the effectiveness of this framework on real world data.

5.1 Framework

In this section we describe the three agents Bob, Alice and Eve, and formulate a camouflage optimization problem for Alice, parametrized by Bob and Eve's definitions.

Table 5.1 Information available to different agents

Agent	**Secret Set,** D_S	**Camouflage Pool,** $\mathscr{C}$	**Bob's Learner,** $\mathscr{A}$	**Detection Function,** Ψ	**Camouflaged Training Set,** $\mathscr{O}$
Bob	No	Yes/No	Yes	Yes/No	Yes
Alice	Yes	Yes	Yes	Yes	Yes
Eve	No	Yes	Yes	Yes	Yes

The agent Bob uses a standard learning algorithm $\mathscr{A} : \mathscr{D} \mapsto \mathscr{H}$ which, given a training set D, learns a hypothesis $\mathscr{A}(D)$ in a hypothesis space $\mathscr{H}$. The resulting hypothesis maps instances in the input space $\mathscr{X}$ to the output space $\mathscr{Y}$. This can be multi-class classification or regression, though in the present work we focus on binary classification. We assume that Bob's learning algorithm is "open source". That is, all information about $\mathscr{A}$ is known to all agents. However, Bob and Alice have shared knowledge on class naming: which class is positive and which negative. For K-class classification this shared knowledge requires $\Theta(K \log K)$ bits.

Alice is the attacker. She has a secret classification task and the corresponding private dataset D_S. In addition, she has access to a public pool of n instances $\mathscr{C} = \{(\mathbf{x}_i, y_i)_{1:n}\}$ (*camouflage pool*) drawn i.i.d. from $\mathbb{Q}_{(\mathbf{x},y)}$ which we call the *cover data distribution*. Note that this is not the distribution from which D_S is drawn. In the preceding example, $\mathbb{Q}_{(\mathbf{x},y)}$ is the distribution over handwritten digits, whereas D_S is a collection of photographs of men and women.

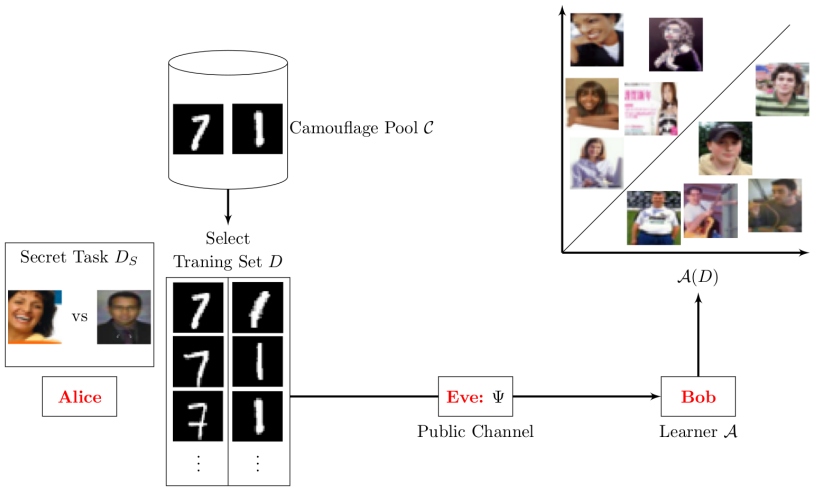


Fig. 5.2 Training set camouflage framework. We show the three agents along with the classification task, camouflage pool, camouflage training set and Eve's detection function

Alice seeks to select a camouflaged training set $D \subset \mathscr{C}$ which she will send to Bob for training. Alice wants Bob to succeed on the secret task, hence a good D should minimize the risk:

$$\mathscr{L}_{\mathscr{A}}(D) = \frac{1}{|D_S|} \sum_{(\tilde{\mathbf{x}},\tilde{y}) \in D_S} \ell(\mathscr{A}(D)(\tilde{\mathbf{x}}), \tilde{y})$$

where $\ell(\hat{y}, y)$ is a loss function. Note the private D_S is used to define the quality of D. Also keep in mind that D_S and D are in different domains.

Eve is a passive eavesdropper. She observes the dataset D which Alice sends to Bob, but does not alter it. Her goal is to determine if D is normal or suspicious. Specifically, Eve is suspicious of any data unless it appears to be an i.i.d. sample drawn from $\mathbb{Q}_{(\mathbf{x},y)}$. She has access to the camouflage pool $\mathscr{C}$, but like Alice she cannot draw additional instances from $\mathbb{Q}_{(\mathbf{x},y)}$. Eve uses $\mathscr{C}$ and a detection function $\Psi : \mathscr{D} \times \mathscr{D} \mapsto \mathbb{R}_+$ to examine D. She flags D as suspicious if $\Psi(\mathscr{C}, D) \geq 0$. As with Bob, we assume that Eve is "open source". Namely, Alice can see Eve's detection function Ψ.

The agents Alice and Eve have conflicting goals. Alice aims to secretly communicate to Bob by way of constructing a training set, while Eve is suspicious and will flag any data sent to Bob if it looks suspect. Bob dutifully learns a model and for our purposes has no explicit intentions. We make explicit each Agent's knowledge in Table 5.1.

With the agents defined, we can now formulate Alice's goal:

$$\begin{aligned} &\operatorname*{argmin}_{D\subset\mathscr{C}} \frac{1}{|D_S|}\sum_{(\tilde{\mathbf{x}},\tilde{y})\in D_S} \ell(\mathscr{A}(D)(\tilde{\mathbf{x}}),\tilde{y}) \\ &\text{s.t. } \Psi(\mathscr{C},D) < 0 \end{aligned} \tag{5.1}$$

That is, she seeks a camouflage training set D from the cover data pool. D should not be flagged as suspicious by Eve. D should also make Bob learn well, comparable to directly giving Bob her private data set D_S. An example of the training set camouflage in action is shown in Figure 5.2.

5.2 MMD as Detection Function

One critical component of our camouflage framework is Eve's detection function Ψ — how she determines if a training set is suspicious or not. Eve's detection function is a two-sample test as its goal is to discern if the two sets $\mathscr{C},D$ are drawn from the same distribution or not. In what follows we derive new results on Maximum Mean Discrepancy (**MMD**) [53] as the detection function as it is a widely used two-sample test [35]. Of course, other detection functions can be used in (5.1).

We first review basic **MMD** following [53]. Let p and p' be two Borel probability measures defined on a topological space $\mathscr{Z}$. Given a class of functions $\mathscr{F}$ such that $f : \mathscr{Z} \mapsto \mathbb{R}, f \in \mathscr{F}$, **MMD** is defined as

$$\mathbf{MMD}(p,p') = \sup_{f\in\mathscr{F}} \left(E_{\mathbf{z}}[f(\mathbf{z})] - E_{\mathbf{z}'}[f(\mathbf{z}')]\right) \tag{5.2}$$

Any unit ball in a reproducing kernel Hilbert space (RKHS) can be used as the function class $\mathscr{F}$ if the kernel is universal (e.g., Gaussian and Laplace kernels [141]). Using this function space, **MMD** is a metric. This means $\mathbf{MMD}(p,p') = 0 \Leftrightarrow p = p'$.

This requires the expectations to be known, which is not available in practice. We obtain an empirical estimation by replacing the population expectations with empirical mean computed on i.i.d. samples $Z = \{\mathbf{z}_1,\ldots,\mathbf{z}_n\}$ and $Z' = \{\mathbf{z}'_1,\ldots,\mathbf{z}'_m\}$ from p and p', respectively. Define $\mathbf{MMD}_b(Z,Z')$ as follows:

$$\left[\frac{1}{n^2}\sum_{i,j=1}^{n} k(\mathbf{z}_i,\mathbf{z}_j) - \frac{2}{nm}\sum_{i,j=1}^{n,m} k(\mathbf{z}_i,\mathbf{z}'_j) + \frac{1}{m^2}\sum_{i,j=1}^{m} k(\mathbf{z}'_i,\mathbf{z}'_j)\right]^{\frac{1}{2}}$$

where k is the kernel of the RKHS. Let $d := |\mathbf{MMD}_b(Z,Z') - \mathbf{MMD}(p,p')|$. Gretton *et. al.* show that

$$P\left(d > 2\left(\sqrt{\frac{K}{n}} + \sqrt{\frac{K}{m}}\right) + \varepsilon\right) \leq 2e^{-\frac{\varepsilon^2 nm}{2K(n+m)}} \tag{5.3}$$

where K is an upperbound on the kernel values.

We convert the above bound into a one-sided hypothesis testing procedure. Under the null hypothesis $p = p'$ we have $\mathbf{MMD}(p,p') = 0$. We consider positive deviations of $\mathbf{MMD}_b(Z,Z')$ from $\mathbf{MMD}(p,p')$. Equating the RHS with α (probability of type I error) gives a hypothesis test of level-α, where solving ε as a function of α gives $\varepsilon = \sqrt{\frac{2K(n+m)}{nm}\log\frac{1}{\alpha}}$. We retain the null hypothesis if

$$\mathbf{MMD}_b(Z,Z') - T < 0 \tag{5.4}$$

where the threshold is

$$T = 2\left(\sqrt{\frac{K}{n}} + \sqrt{\frac{K}{m}}\right) + \sqrt{\frac{2K(n+m)}{nm}\log\frac{1}{\alpha}}. \tag{5.5}$$

This also defines Eve's detection function ($\Psi(\mathscr{C},D)$) at level-α:

$$\Psi(\mathscr{C},D) \equiv \mathbf{MMD}_b(\mathscr{C},D) - T. \tag{5.6}$$

If $\Psi(\mathscr{C},D) \geq 0$ then Eve realizes that D is not drawn i.i.d. from $\mathbb{Q}_{(\mathbf{x},y)}$ and flags it as suspicious.

We now have a fully specified training set camouflage optimization problem (5.1).

5.3 Solving the Optimization Problem

In this section, we propose two different solvers for the optimization problem defined in (5.1). We first show how the optimization problem can be reduced to a nonlinear programming problem for some specific learners. This approach is only applicable to convex learners. So, in addition we present a combinatorial search method as a generic heuristic solver.

5.4 Nonlinear Programming (NLP)

We assume Bob's machine learning algorithm $\mathscr{A}$ is convex, specifically regularized empirical risk minimization. This covers a wide range of learners such as support vector machines, logistic regression, and ridge regression. Let Θ be Bob's hypothesis space, ℓ his loss function, and λ his regularization parameter, respectively. Let $m := |D|$ be given. We convert Alice's optimization problem (5.1) into nonlinear programming in the following steps.

Step 1. Using the definition of Bob, we rewrite (5.1) as

$$\begin{aligned} \min_{D\subset\mathscr{C},\hat{\theta}} \quad & \frac{1}{|D_S|}\sum_{(\tilde{\mathbf{x}},\tilde{y})\in D_S}\ell(\hat{\theta},\tilde{\mathbf{x}},\tilde{y}) \\ \text{s.t.} \quad & \hat{\theta} = \operatorname*{argmin}_{\theta\in\Theta}\sum_{(\mathbf{x},y)\in D}\ell(\theta,\mathbf{x},y)+\frac{\lambda}{2}\|\theta\|^2 \\ & \Psi(\mathscr{C},D) < 0, \\ & |D| = m. \end{aligned} \tag{5.7}$$

This is still a difficult combinatorial bilevel problem.

Step 2. Since Bob's learning problem is assumed to be convex, its Karush-Kuhn-Tucker (KKT) conditions are sufficient and necessary [160, 84]. We replace the lower level optimization problem in (5.7) with the KKT conditions:

$$\begin{aligned} \min_{\theta\in\Theta,D\subset\mathscr{C}} \quad & \frac{1}{|D_S|}\sum_{(\tilde{\mathbf{x}},\tilde{y})\in D_S}\ell(\hat{\theta},\tilde{\mathbf{x}},\tilde{y}) \\ \text{s.t.} \quad & \sum_{(\mathbf{x},y)\in D}\nabla\ell(\theta,\mathbf{x},y)+\lambda\theta = 0, \\ & \Psi(\mathscr{C},D) < 0, \\ & |D| = m. \end{aligned} \tag{5.8}$$

This is still a combinatorial problem.

Step 3. We introduce binary indicator variable b_i for each instance $(\mathbf{x}_i, y_i) \in \mathscr{C}$. A value of 1 indicates that the instance is a member of the training set D. This yields:

$$\begin{aligned}
\min_{\theta\in\Theta; b_1,\ldots,b_{|\mathscr{C}|}; b_i\in\{0,1\}} \quad & \frac{1}{|D_S|} \sum_{(\tilde{\mathbf{x}},\tilde{y})\in D_S} \ell(\theta, \tilde{\mathbf{x}}, \tilde{y}) \\
\text{s.t.} \quad & \sum_{i=1}^{n} b_i \nabla \ell(\theta, \mathbf{x}_i, y_i) + \lambda\theta = 0, \\
& \Psi(\mathscr{C}, \{b_i(\mathbf{x}_i, y_i) | (\mathbf{x}_i, y_i) \in \mathscr{C}\}) < 0 \qquad (5.9) \\
& \sum_{i=1}^{n} b_i = m. \qquad (5.10)
\end{aligned}$$

This is known as a Mixed Integer Non-Linear Optimization Problem (MINLP). MINLP problems are generally hard to solve in practice. Thus we then relax b_i to be continuous in $[0,1]$, resulting in the following NLP problem:

$$\begin{aligned}
\min_{\theta\in\Theta; b_1,\ldots,b_n\in[0,1]} \quad & \frac{1}{|D_S|} \sum_{(\tilde{\mathbf{x}},\tilde{y})\in D_S} \ell(\theta, \tilde{\mathbf{x}}, \tilde{y}) \\
\text{s.t.} \quad & \sum_{i=1}^{n} b_i \nabla \ell(\theta, \mathbf{x}_i, y_i) + \lambda\theta = 0, \\
& \Psi(\mathscr{C}, \{b_i(\mathbf{x}_i, y_i) | (\mathbf{x}_i, y_i) \in \mathscr{C}\}) < 0, \\
& \sum_{i=1}^{n} b_i = m. \qquad (5.11)
\end{aligned}$$

Note that in this equation we scale the gradient of the loss function for each $(\mathbf{x}_i, y_i)$ by the corresponding b_i. This b_i indicates the importance of an instance in the training set. This is different from training the learner on the complete camouflage pool or one of its subset. Also when calculating the detection function we weigh each instance in the training set by its corresponding b_i. After solving this continuous optimization problem we round the solution by setting the top m values to 1, and the rest to 0. These indices form the camouflaged training set D.

5.5 Beam Search

For Bob with a general learner that cannot be formulated as convex empirical risk minimization, we now describe a heuristic beam search algorithm [118] that directly solves Alice's optimization problem (5.1). Algorithm 2 shows the search procedure. The state space

consists of all training sets $\mathscr{O} \subset \mathscr{C}$ such that $|\mathscr{O}| = m$ and $\Psi(\mathscr{C}, \mathscr{O}) < 0$. Two training sets that differ by one instance are deemed as neighbors. The beam is initialized by selecting training sets at random. The width (w) of the beam is fixed beforehand. For computational efficiency, in each step of beam search we evaluate a randomly selected subset of neighbors for each training set in the beam. This makes the search stochastic in nature. From the union of evaluated neighbors and training sets in the current beam, we select the top w training sets (based on the value of the objective function in (5.1)) to reinitialize the beam and discard the rest. We repeat this process until a pre-specified search budget (number of times $\mathscr{A}$ is trained) is met.

Algorithm 2 Beam Search for Solving the Camouflage Problem

1: Input: Camouflage Pool: $\mathscr{C}$, Risk: $\mathscr{L}_{\mathscr{A}}$, Beam Width: w, Budget: B, Neighborhood Function: $\mathscr{N}$, Size: m, Detection Function: Ψ
2: $\mathscr{D} \leftarrow w$ randomly selected subsets of size m from $\mathscr{C}$ such that $\Psi(\mathscr{C}, D) < 0$
3: **for** $t = 1 \rightarrow B$ **do**
4: $\quad \mathscr{D} \leftarrow \mathscr{D} \cup \mathscr{N}(\mathscr{D}, \mathscr{C}, \Psi)$
5: $\quad \mathscr{D} \leftarrow w$ training sets from $\mathscr{D}$ with smallest $\mathscr{L}_{\mathscr{A}}(D)$ values
6: **end for**
7: **return** $\operatorname{argmin}_{D \in \mathscr{D}} \mathscr{L}_{\mathscr{A}}(D)$

5.6 Experiments

We investigated the effectiveness of training set camouflage through empirical experiments on real world datasets. Our results show that camouflage works on a variety of image and text classification tasks: Bob can perform well on the secret task after training on the camouflaged training set, and the camouflaged training set passes Eve's statistical test undetected.

Bob. We considered two different learning algorithms for Bob: logistic regression and artificial neural network (ANN). For the logistic regression learner, the weight of the regularization parameter was set to 1. The neural network had one hidden layer containing 100 ReLU units [163], and a single output unit with TanH [66] activation function. The learning rate and number of epochs were set to 0.001 and 10 respectively.

Eve. The level-α for Eve's **MMD** detection function was set to 0.05 (i.e., 95% confidence). For all our experiments Eve used the RBF kernel $k(\mathbf{z}_i, \mathbf{z}_j) = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{2\sigma^2}\right)$. Eve set σ to be the median distance between points in the camouflage pool as proposed in [53]. Eve also included the scaled class label as a feature dimension: $[\mathbf{x}_i, c\mathbb{1}\{y_i = 1\}]$ where $c = \max_{k,l \text{ such that } y_k = y_l} \|\mathbf{x}_k - \mathbf{x}_l\|$ and $\mathbb{1}\{\cdot\}$ is the indicator function. This augmented feature enables Eve to monitor both features and labels.

Alice. For the ANN Bob, Alice only used beam search (nonconvexity prevents application of KKT); while for logistic regression she used both solvers. For beam search, the search budget was set to $200,000$, then repeated five times with random restarts. The loss ℓ was logistic loss and squared loss for logistic regression and ANN, respectively. Alice constructed camouflaged training sets of size $m = 20$, 100 and 500.

Evaluation metrics. As is standard to estimate general performance, we used a separate test set, generated from the same distribution as the secret set D_S and not known to any agent, to evaluate Bob's generalization error when trained on Alice's camouflaged training set D. We compare this to two quantities: ("random") when Bob is trained on a uniform sample of size m from the cover data distribution, which we expect to perform poorly; and ("oracle") when Bob is trained directly on Alice's secret set D_S, ignoring Eve's presence. The oracle gives us an estimate on how much performance Bob is losing due to camouflage.

5.6.1 Datasets

We performed eight camouflage experiments: WM71, WMOA, GP52, GPOA, CABH, CAIM, DRBH and DRIM as described below. The first two letters in the acronyms encode the secret task, and the last two the cover task. In both WM71 and WMOA the secret task was woman vs. man (CIFAR-100), the cover task was handwritten digits 7 vs. 1 (MNIST) and orange vs. apple (CIFAR-100), respectively. In GP51 and GPOA the secret task was handgun vs. phone image classification (OpenImages [74]), the cover was 5 vs 1 and orange vs. apple, respective. In CABH and CAIM the secret task was text classification on christian vs atheism (20-newsgroups), the cover tasks were baseball vs hockey and pc vs mac, respectively. In DRBH and DRIM the secret task was text classification on Democratic vs. Republican (All The News dataset [147]), the cover tasks were baseball vs hockey and pc vs mac, again. For all datasets, the positive/negative examples of the sensitive set were mapped to the positive/negative examples of the camouflage pool. We do so because we assume that Alice is not allowed to modify the camouflage pool.

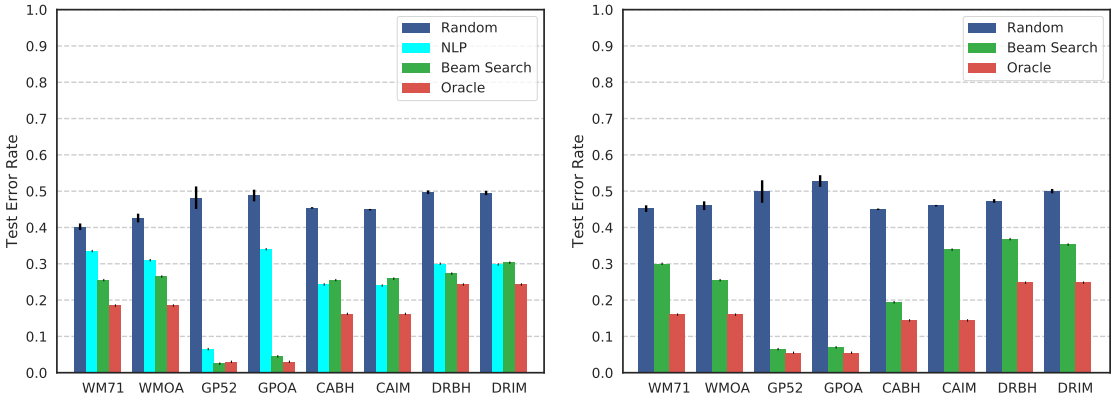
For images we used ResNet [57] to generate feature vectors of dimension 2048. For text we used Word2Vec [97] to generate feature vectors of dimension 300 by averaging over the word vectors in an article. A summary of the datasets can be found in Table 5.2.

5.6.2 Results

We present our results in Figure 5.3. For brevity, we only show $m = 20$ and $m = 500$ results respectively for the image and text experiments. But we note that the results presented herein are representative of all experiments performed.

Table 5.2 Dataset Summary

Name	Type	# Features	# Camouflage Pool Positive	# Camouflage Pool Negative	# Secret Positive	# Secret Negative	# Test Positive	# Test Negative
WM71	Image	2048	600	600	500	500	100	100
WMOA	Image	2048	600	600	500	500	100	100
GP52	Image	2048	600	600	400	400	100	100
GPOA	Image	2048	600	600	400	400	100	100
CABH	Text	300	994	999	599	480	398	319
CAIM	Text	300	982	963	599	480	398	319
DRBH	Text	300	994	999	800	800	200	200
DRIM	Text	300	982	963	800	800	200	200



(a) Logistic Regression Learner

(b) Artificial Neural Network Learner

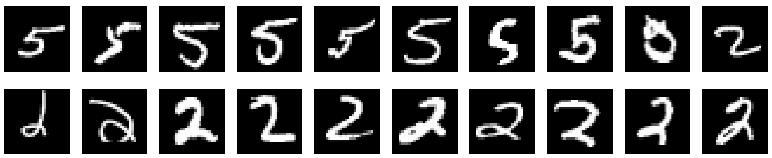
Fig. 5.3 Test error rates found by solving the camouflage framework. We also show random (with error bars) and oracle error for comparison. For image and text datasets we show results for $m = 20$ and $m = 500$ respectively for brevity.

We observe that Alice, using either NLP or beam search solver, can find much better camouflage training sets than random and in many cases approach oracle. Keep in mind that Alice's solutions do not trigger Eve's suspicion function.

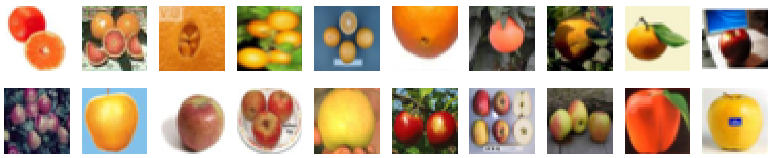
Figures 5.4, 5.5 visualize results on image experiments WM71, WMOA, GP52, GPOA when Bob's learner is logistic regression. Visually, the camouflaged training set D bears no resemblance to the sensitive training set D_S. This is true for text camouflage experiments as well, where articles in the camouflaged training sets have no obvious semantic connection to the sensitive task. See Table 5.3 for results on the text experiment CABH. This is indeed bad news for human Eves: not only did camouflage fooled MMD detector, it will also likely fool human inspectors.

We also observe that beam search largely outperforms NLP. If we performed MINLP instead of NLP then we would have expected MINLP to find better solutions than beam search. This suggests that our simple relaxation and rounding strategy in NLP needs improvement. It

(a) Gun vs. phone, the secret task in GP52 and GPOA



(b) Camouflaged training set using 5 vs. 2 (GP52)



(c) Camouflaged training set using oranges vs. apples (GPOA)

Fig. 5.4 Camouflage results for GP52 and GPOA experiments

should also be noted that the solutions found by NLP may not be feasible (i.e., $\Psi(\mathscr{C},\mathscr{O}) > 0$). But this scenario did not occur during our experiments. All these observations combined with the fact that beam search is also applicable to a wider range of learners, makes beam search the preferred solver for Alice.

We note that **MMD** is stronger with larger sample sizes. It will be harder for Alice to fool Eve given a large camouflage pool C and also if she is forced to select a large camouflaged training set D. But this also depends on Bob's learning algorithm as well. If Bob is using support vector machines (SVM) then a large sample size requirement should not be an issue. Alice only needs to focus on the support vectors in such a scenario and can then choose the remaining samples randomly. **MMD** is also stronger with smaller feature dimensions [53].

We designed the experiments so that Alice needs to teach the same secret task (e.g. woman vs. man) to Bob using two different cover tasks (e.g. 7 vs 1, and orange vs. apple). Interestingly, which cover task was used does not matter much. Conversely, Alice also needs to teach different secret tasks (e.g. woman vs. man, and gun vs. phone) to Bob using the same cover task (e.g. orange vs. apple). Alice also succeeds to some degree in both cases.

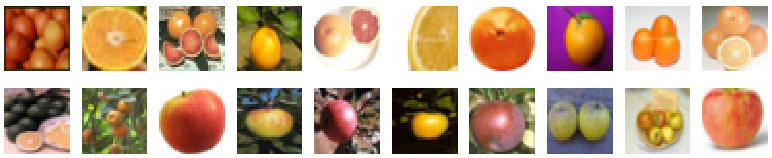


Fig. 5.5 Camouflaged training set using oranges vs. apples (WMOA). The secret task woman vs. man was shown in Figure 5.1b, while another camouflage training set using 7 vs. 1 was shown in Figure 5.1a.

Table 5.3 Camouflage results for the CABH experiment

Sample of Sensitive Set		**Sample of Camouflaged Training Set**	
Class	**Article**	**Class**	**Article**
Christianity	...Christ that often causes christians to be very critical of themselves and other christinas. We...	Baseball	...The Angels won their home opener against the Brewers today before 33,000+ at Anaheim Stadium...
	...I've heard it said that the accounts we have of Christs life and ministry in the Gospels were...		... interested in finding out how I might be able to get two tickets for the All Star game in Baltimore...
Atheism	...This article attempts to provide a general introduction to atheism. Whilst I have tried to be...	Hockey	... user and not necessarily those could anyone post the game summary for the Sabres-Bruins game....
	...Science is wonderful at answering most of our questions. I'm not the type to question scientific...		...Tuesday, and the isles/caps game is going into overtime. what does ESPN do. Tom Mees says, "we...

We note that camouflage seems easier for Alice to do if the cover task is in fact somewhat confusable, presumably because she can generate different decision boundaries by picking from overlapping camouflage items. One interesting open question is whether there is a *universal* cover task for all secret tasks (they must use the same feature representation, of course).

As mentioned previously, Bob fixed his learning hyperparameters. This was done for speed. However, nothing prevents Bob from tuning his hyperparameters by cross validation. Alice would simply emulate the same cross validation while optimizing the camouflaged training set. This can be easily done in beam search, at the cost of more computation.

Also, the loss function ℓ used by Alice and Bob is the same, as seen in the upper and lower optimization problems in (5.7). It is straightforward to allow different losses. For example, Bob may learn with the logistic loss since it is a standard learner, while Alice may use 0-1 loss to directly optimize Bob's accuracy.

Moreover, we note that the assumption of Eve's detection function being known to Alice can be relaxed as well. If Alice does not have access to said detection function then it is possible for her to generate a surrogate of that function by probing Eve with multiple datasets (while not training Bob). Once Alice is confident about the surrogate function, she can again apply the same techniques to generate the camouflaged training set. We leave out this discussion for future work.

5.7 Contribution

My main contribution in this project is to help formulating the training set camouflage problem. Moreover, I helped devise the different approaches to solve the optimization problem. Finally, I also ran the experiments and helped to analyze the data. Scott Alfeld, Computer Science, Amherst College also helped us in formulating the training set camouflage problem presented in this work. The work presented in this chapter has been published at "Ayon Sen, Scott Alfeld, Xuezhou Zhang, Ara Vartanian, Yuzhe Ma, and Xiaojin Zhu. 'Training set camouflage.' International Conference on Decision and Game Theory for Security (2018)."

Chapter 6

Verifying Visual Imperceptible Measures in Adversarial Attacks

Adversarial test-time attacks perturb an input item $\mathbf{x}_0$ slightly into $\mathbf{x}$ such that (1) $\mathbf{x}$ is classified differently than $\mathbf{x}_0$; (2) the change from $\mathbf{x}_0$ to $\mathbf{x}$ is small. The oft-quoted reason for (2) is to make the attack hard to detect [145, 52, 98, 24]. This assumes an inspector, who detects suspicious items before sending them to the classifier [125]. We focus on image classification where the inspector is an "ideal observer" who possesses population median human perception. For example, one popular measure is the pixel p-norm, which assumes that perturbations lying inside the norm ball $\{\mathbf{x} : \|\mathbf{x}-\mathbf{x}_0\|_p \leq r\}$ are imperceptible to humans. Here we define the pixel feature space as $\mathscr{X} \in \mathbb{R}^d$ with the understanding that normal pixel values are integers between 0 and 255. While many researchers readily acknowledge that pixel p-norm is only a surrogate to attack detectability, **the scientific community has yet to quantitatively understand how effective this or other surrogates are**. Our work answers the question: How well do pixel p-norm or other popular measures match human inspectors?

To see why a potential mismatch matters, let us consider pixel p-norm as an example. Figure 6.1 shows a hypothetical image space around input $\mathbf{x}_0$. Human imperceptibility region is depicted by green: an image in this region looks like $\mathbf{x}_0$ to an average human. The pixel p-norm ball with the optimal norm p and radius r is shown in gray. The mismatch prevents adversarial machine learning from accurately studying the effect of attacks. For instance, the attacker may give up using $\mathbf{x}_2$ as an attack because $\|\mathbf{x}_2-\mathbf{x}_0\|_p$ is large, but in reality $\mathbf{x}_2$ can indeed pass the human inspector. This is equivalent to a type I error (false positive) in statistical hypothesis testing. Conversely, attacking with $\mathbf{x}_1$ is futile because the human inspector readily detects the attack even though $\|\mathbf{x}_1-\mathbf{x}_0\|_p$ is small. This is equivalent to a type II error (false negative).

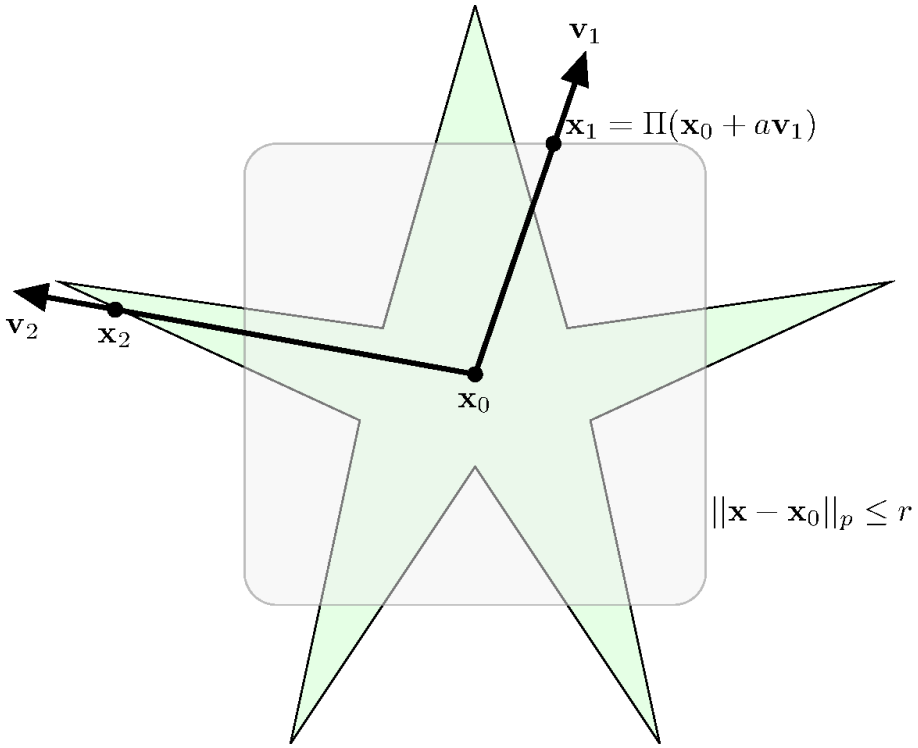


Fig. 6.1 Schematic diagram of mismatch between human perception and pixel p-norm

We formally define a perceptibility detector by a distance $\rho : \mathscr{X} \times \mathscr{X} \to \mathbb{R}_{\geq 0}$ and threshold r, such that the detector flags a perturbed image $\mathbf{x}$ with respect to an original image $\mathbf{x}_0$ if $\rho(\mathbf{x}, \mathbf{x}_0) \geq r$. ρ can be any computable distance such as pixel p-norm, earth mover's distance, etc. and does not need to be a metric. The ideal human observer determines the true perceptibility of $\mathbf{x}$ w.r.t. $\mathbf{x}_0$.

Definition 1 (Type I error) *The ideal observer perceives* $\mathbf{x}_0$ *and* $\mathbf{x}$ *to be the same but* $\rho(\mathbf{x}, \mathbf{x}_0) \geq r$.

Definition 2 (Type II error) *The ideal observe notices the difference between* $\mathbf{x}_0$ *and* $\mathbf{x}$ *but* $\rho(\mathbf{x}, \mathbf{x}_0) < r$.

Both types of error have occurred in practice, as presented in Figures 6.2 and 6.3 respectively. In both examples we used $\rho(\mathbf{x}, \mathbf{x}_0) = \|\mathbf{x} - \mathbf{x}_0\|_\infty$ and $r = 8$, as is commonly used in adversarial machine learning [6, 164].

While the concept of perceptibility mismatch is well-known, it is not clear how popular distances compare to human perception. Out of 32 recent papers we surveyed, 27 papers (each with over 100 citations) used pixel p-norms in attacks. Among these 27, 20% assumed p-norms are a good match to human perception without providing evidence; 50% used them

because other papers did; and the rest used them without justification. Hence, a large portion of adversarial machine learning research is based on unknown ground.

Gaining human perception knowledge requires multiple interdisciplinary studies in adversarial machine learning and cognitive science. The seminal work by Sharif *et al.* performed a behavioral study on adversarial attack and human perception [133]. They showed that humans may categorize two perturbed thumbnails – of the same pixel p-norm (for $p = 0, 2, \infty$) distance to the original thumbnail – differently. While valuable, their conclusions are limited due to the study design: they only tested pixel 0-, 2-, ∞-norms but not other p-norms or measures. Their test also required knowledge of the radius r, and depended on humans (mis)-categorizing a low resolution thumbnail (MNIST [83], CIFAR10 [77]), which does not reflect humans' ability to notice small changes in a normal-sized image well before humans' categorization on that image changes.

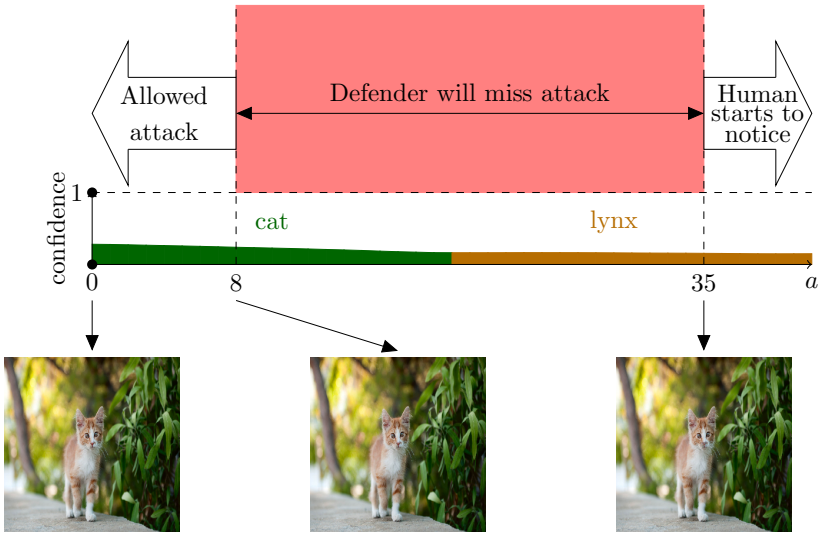


Fig. 6.2 Example of a type I error: defender potentially missing an attack in a specific direction (M_RGB_Box). Details about the direction can be found in Section 6.2

.

Our work significantly extends and complements [133], and addresses all these issues: Our design enables us to test all pixel p-norms along with other popular measures like earth mover's distance, structural similarity (SSIM), and deep neural network representation. It is also agnostic to the true value of r by using the notion of human just-noticeable-difference. We test humans in small image-change regimes that better match what a human inspector typically faces in an adversarial setting. **Our main results caution against the use of pixel p-norms, earth mover's distance, structural similarity, or deep neural network**

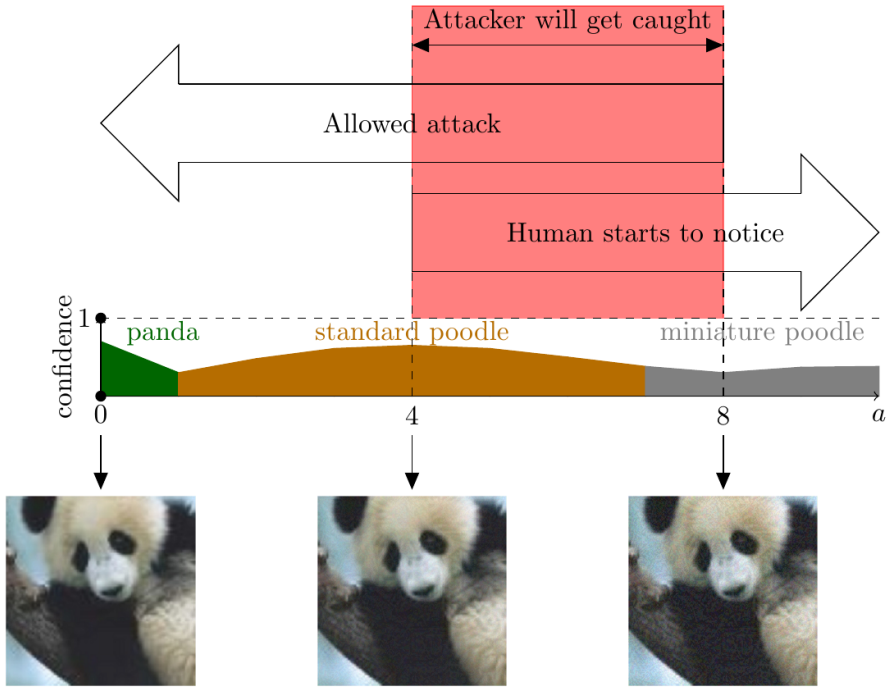


Fig. 6.3 Example of a type II error: an attack produced by FGSM that will get caught.

representation by quantifying their match to human judgment. These results have the potential to improve the understanding of adversarial attack and defense strategies.

We also mention some limitations of our work. We cannot directly answer "what is the correct measure", because computationally modeling human visual perception is still an open question in psychology [138, 116, 158, 61]. We used a "show $\mathbf{x}_0$ then perturb" experiment paradigm, while in real applications the human inspector may not have access to $\mathbf{x}_0$. We also limit ourselves to the visual domain. These topics remain future work.

6.1 The Central Hypothesis and Its Implications

Consider a natural image $\mathbf{x}_0 \in \mathscr{X}$ and another image $\mathbf{x}$. The central hypothesis is the following.

Definition 3 (The Central Hypothesis) *A measure $\rho : \mathscr{X}^d \times \mathscr{X}^d \to \mathbb{R}_{\geq 0}$ is a good fit for adversarial machine learning research if $\forall \mathbf{x}_0, \exists$ threshold $r(\mathbf{x}_0)$ such that the ideal observer perceives any $\mathbf{x}$ the same as $\mathbf{x}_0$ if $\rho(\mathbf{x}, \mathbf{x}_0) < r(\mathbf{x}_0)$, and the ideal observer notices the difference if $\rho(\mathbf{x}, \mathbf{x}_0) \geq r(\mathbf{x}_0)$.*

The threshold $r(\mathbf{x}_0)$ is known as the "Just Noticeable Difference" (JND) in experimental psychology [41, 166]. We further define the set of Just-Noticeably-Different images with

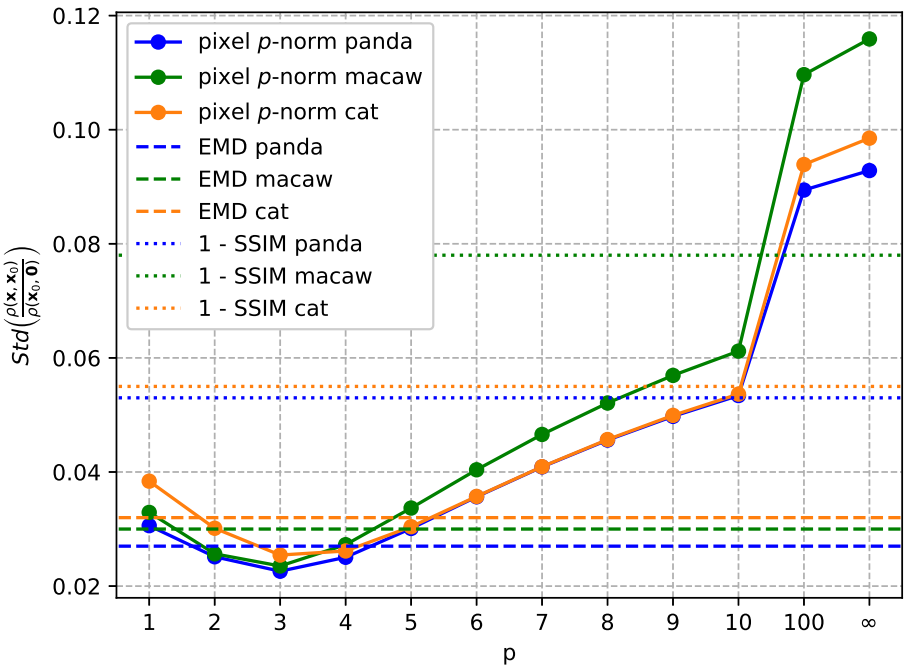


Fig. 6.4 Variability of fit to human data, lower is better. See section 6.4 for details and discussion

respect to $\mathbf{x}_0$ under the central hypothesis: $J(\mathbf{x}_0) \coloneqq \{\mathbf{x} \subset \mathscr{X} : \rho(\mathbf{x}, \mathbf{x}_0) = r(\mathbf{x}_0)\}$. In other words, $J(\mathbf{x}_0)$ is the shell of the ball (as defined by the measure $\rho(*,*)$) centered at $\mathbf{x}_0$ with radius $r(\mathbf{x}_0)$. One of the primary tasks of the present paper is to test if the central hypothesis holds true for some popular measures used in practice. These measures are defined as follows.

Pixel p-norm. For any $p \in [0,\infty]$ it measures the amount of perturbation by $\|\mathbf{x}-\mathbf{x}_0\|_p \coloneqq \left(\sum_{i=1}^{d} |x_i - x_{0,i}|^p\right)^{1/p}$. We define the 0-norm to be the number of nonzero elements. We test $\rho(\mathbf{x}, \mathbf{x}_0) = \|\mathbf{x}-\mathbf{x}_0\|_p$ for any $p \in [0,\infty]$.

Table 6.1 Attributes of ± 1-perturbation directions

# Dimensions Changed (s)	S = 1, M = 288 L = 30603, X = 268203 (mnemonic: garment size)
Color Channels Affected	Red = only the red channel of a pixel RGB = all three channels of a pixel
Shape of Perturbed Pixels	Box = a centered rectangle Dot = scattered random dots Eye = on the eye of the animal

Earth mover's distance (EMD). Also known as Wasserstein distance, it is a distance function defined between two probability distributions on a given metric space. The metric computes the minimum cost of converting one distribution to the other one. EMD has been used as a distance metric in the image space also, e.g. for image retrieval [124]. Given two images $\mathbf{x}_0$ and $\mathbf{x}$, EMD is calculated as $EMD(\mathbf{x}_0,\mathbf{x}) = \inf_{\gamma\in\Gamma(\mathbf{x}_0,\mathbf{x})} \int_{\mathbb{R}\times\mathbb{R}} |a-b| d\gamma(a,b)$. Here, $\Gamma(\mathbf{x}_0,\mathbf{x})$ is the set of joint distributions whose marginals are $\mathbf{x}_0$ and $\mathbf{x}$ (treated as histograms), respectively. We test $\rho(\mathbf{x},\mathbf{x}_0) \coloneqq EMD(\mathbf{x},\mathbf{x}_0)$ which assumes that the same amount of earth moving in images corresponds to the same detectability by human perception.

Structural Similarity (SSIM). This measure is intended to be a perceptual similarity measure that quantifies image quality loss due to compression [156], and used as a signal fidelity measure with respect to humans in multiple research works [155, 134]. SSIM has three elements: luminance, contrast and similarity of local structure. Given two images $\mathbf{x}_0$ and $\mathbf{x}$, SSIM is defined by $SSIM(\mathbf{x}_0,\mathbf{x}) = \left(\frac{2\mu_{\mathbf{x}_0}\mu_{\mathbf{x}}+C_1}{\mu_{\mathbf{x}_0}^2+\mu_{\mathbf{x}}^2+C_1}\right)\left(\frac{2\sigma_{\mathbf{x}_0}\sigma_{\mathbf{x}}+C_2}{\sigma_{\mathbf{x}_0}^2+\sigma_{\mathbf{x}}^2+C_2}\right)\left(\frac{\sigma_{\mathbf{x}_0\mathbf{x}}+C_3}{\sigma_{\mathbf{x}_0}\sigma_{\mathbf{x}}+C_3}\right)$. $\mu_{\mathbf{x}_0}$ and $\mu_{\mathbf{x}}$ are the sample means; $\sigma_{\mathbf{x}_0}$, $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{x}_0\mathbf{x}}$ are the standard deviation and sample cross correlation of $\mathbf{x}_0$ and $\mathbf{x}$ (after subtracting the mean) respectively. To compute SSIM we use window size 7 without Gaussian weights. Since SSIM is a similarity score, we define $\rho(\mathbf{x},\mathbf{x}_0) = 1 - SSIM(\mathbf{x},\mathbf{x}_0)$.

Deep neural network (DNN) representation. Even though DNNs are designed with engineering goals in mind, studies comparing their internal representations to primate brains have found similarities [76]. Let $\xi(\mathbf{x}) \in \mathbb{R}^D$ denote the last hidden layer representation of input image $\mathbf{x}$ in a DNN. We define $\rho(\mathbf{x},\mathbf{x}_0) = \|\xi(\mathbf{x}) - \xi(\mathbf{x}_0)\|_p$ as a potential distance metric for our purpose. We use Inception V3 representations with $D = 2048$.

To test the central hypothesis for the aforementioned measures, we derive a number of testable implications. These implications will be tested through human behavioral experiments in later sections. The first implication follows trivially from the definition of $J(\mathbf{x}_0)$. It states that any Just-Noticeably-Different images of an $\mathbf{x}_0$ has the same $\rho(*,*)$ measure (does not require knowledge of $r(\mathbf{x}_0)$):

Implication 1 *Suppose $\rho : \mathscr{X}\times\mathscr{X} \to \mathbb{R}_{\geq 0}$ is a good fit according to the central hypothesis. Then $\forall \mathbf{x}_0, \forall \mathbf{x}_1, \mathbf{x}_2 \in J(\mathbf{x}_0), \rho(\mathbf{x}_1,\mathbf{x}_0) = \rho(\mathbf{x}_2,\mathbf{x}_0)$.*

The second implication, even though only applicable for pixel p-norms, is more powerful in the sense that it can be tested for all $p \in [0,\infty]$ simultaneously without knowing $r(\mathbf{x}_0)$. We introduce this second implication as it is infeasible to test implication 1 for all $p \in [0,\infty]$ in case of pixel p-norm. To test the second implication, we utilize special perturbed $\mathbf{x}$ as follows. As indicated in Figure 6.1, we consider $\mathbf{x}$ generated along the ray defined by a **perturbation direction** $\mathbf{v} \in \mathbb{R}^d$ with a **perturbation scale** $a > 0$: $\mathbf{x} = \Pi(\mathbf{x}_0 + a\mathbf{v})$.

Here Π is the projection onto $\mathscr{X}$; namely, clipping values to $[0,255]$ and rounding to integers. Note that as a increases, the perturbation becomes stronger. The perturbation direction $\mathbf{v}$ is important: in our experiments some directions are generated by popular adversarial attacks in the literature, while others are designed to facilitate statistical tests. Specifically, we define ± 1**-perturbation directions** as any $\mathbf{v} \in \mathbb{R}^d$ with the following two properties: (i) Its support (nonzero elements) has cardinality $s > 0$; in many cases $\mathbf{v}$ will be sparse with $s \ll d$; (ii) the nonzero elements v_i are either 1 or -1 depending on the value of the corresponding element $x_{0,i}$ in $\mathbf{x}_0$: $v_i = 1$ if $x_{0,i} < 128$, and -1 otherwise. For ± 1-perturbations $\mathbf{v}$ and integer $a \in \{1,\ldots,128\}$ it is easy to see that the projection Π is not needed: $\mathbf{x} = \Pi(\mathbf{x}_0 + a\mathbf{v}) = \mathbf{x}_0 + a\mathbf{v}$. This allows for convenient experiment design. More importantly, for such ± 1-perturbed images any pixel p-norm has a simple form: $\forall p : \|\mathbf{x} - \mathbf{x}_0\|_p = \left(\sum_{\mathbf{v}_i \neq 0} |a v_i|^p\right)^{1/p} = a s^{1/p}$. Implication 2 states that two just-noticeable perturbed images with the same perturbation sparsity s should have the same perturbation scale a. Importantly, it can be tested for all $p \in [0,\infty]$ simultaneously without knowing $r(\mathbf{x}_0)$. If it fails then no pixel p-norm is appropriate to model human perceptions of just-noticeable-difference.

Implication 2 *For pixel p-norm,* $\forall p > 0$, $\forall \mathbf{x}_0$, $\forall \pm 1$*-perturbation directions* $\mathbf{v}_1, \mathbf{v}_2$ *with the same sparsity s, suppose* $\exists a_1, a_2 \in \{1,\ldots,128\}$ *such that* $\mathbf{x}_1 = \mathbf{x}_0 + a_1\mathbf{v}_1 \in J(\mathbf{x}_0)$ *and* $\mathbf{x}_2 = \mathbf{x}_0 + a_2\mathbf{v}_2 \in J(\mathbf{x}_0)$. *Then* $a_1 = a_2$.

While implication 2 focuses on perturbation directions of the same sparsity, the next implication states that if one perturbation is changing more dimensions than the other, it should achieve just noticeable difference with a smaller perturbation scale. Again, this is true for all pixel p-norm.

Implication 3 *For pixel p-norm,* $\forall p > 0$, $\forall \pm 1$*-perturbation directions* $\mathbf{v}_1, \mathbf{v}_2$ *with sparsity* $s_1 > s_2$, *suppose* $\exists a_1, a_2$ *such that* $\mathbf{x}_1 = \mathbf{x}_0 + a_1\mathbf{v}_1 \in J(\mathbf{x}_0)$ *and* $\mathbf{x}_2 = \mathbf{x}_0 + a_2\mathbf{v}_2 \in J(\mathbf{x}_0)$. *Then* $a_1 < a_2$.

6.2 Behavioral experiment design

We conducted a human behavioral experiment under Institutional Review Board (IRB) approval. The figures below are best viewed by zooming in to replicate the participant experience. We will release all behavioral data to the public upon publication for reproducibility and further research.

Center images $\mathbf{x}_0$ and perturbation directions $\mathbf{v}$: We chose three natural images (from the Imagenet dataset [34]) popular in adversarial research: a panda [52], a macaw [99]

and a cat [6] as $\mathbf{x}_0$ in our experiment (Figure 6.5). We resized the images to 299×299 to match the input dimension of the Inception V3 image classification network [144]. For each natural image $\mathbf{x}_0$ we considered 10 perturbation directions $\mathbf{v}$, see Figure 6.6. Eight are specially crafted ± 1-perturbation directions varying in three attributes (Table 6.1), and further explained in the caption of Figure 6.6.

(a) panda (b) macaw (c) cat

Fig. 6.5 The three natural images $\mathbf{x}_0$

The remaining two perturbation directions are adversarial directions. We used Fast Gradient Sign Method (FGSM) [52] and Projected Gradient Descent (PGD) [93] to generate two adversarial images $\mathbf{x}^{FGSM}, \mathbf{x}^{PGD}$ for each $\mathbf{x}_0$, with Inception V3 as the victim network. All attack parameters are set as suggested in the methods' respective papers. PGD is a directed attack and requires a target label; we choose gibbon (on panda) and guacamole (on cat) following the papers, and cleaver (on macaw) arbitrarily. We then define the adversarial perturbation directions by $\mathbf{v}^{FGSM} = 127.5(\mathbf{x}^{FGSM} - \mathbf{x}_0)/\|\mathbf{x}^{FGSM} - \mathbf{x}_0\|_2$ and $\mathbf{v}^{PGD} = 127.5(\mathbf{x}^{PGD} - \mathbf{x}_0)/\|\mathbf{x}^{PGD} - \mathbf{x}_0\|_2$. We use the factor 127.5 based on a pilot study to ensure that changes between consecutive images in the adversarial perturbation directions are not too small or too big.

Experimental procedure: See Figure 6.7. Each participant was first presented with instructions and then completed a sequence of 34 trials, of which 30 were ± 1-perturbation or adversarial trials, and 4 were guard trials. The order of these trials was randomized then fixed (see figure). During each trial the participants were presented with an image $\mathbf{x}_0$. They were instructed to increase (decrease) perturbations to this image by using right / left arrow keys or buttons. Moving right (left) incremented (decremented) a by 1, and the subject was then presented with the new perturbed image $\mathbf{x} = \Pi(\mathbf{x}_0 + a\mathbf{v})$. We did not divulge the nature of the perturbations $\mathbf{v}$ beforehand, nor the current perturbation scale a the participant had added to $\mathbf{x}_0$ at any step of the trial. **The participants were instructed to submit the perturbed**

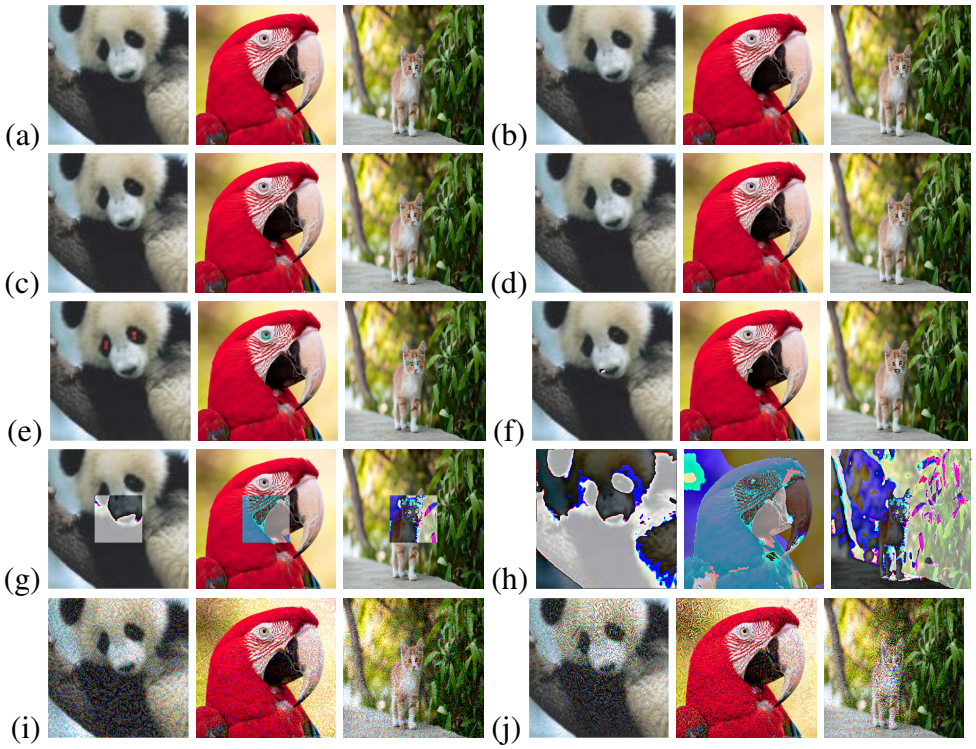


Fig. 6.6 All 10 perturbation directions $\mathbf{v}$ with severe perturbation scale $a = 128$. (a) S_Red_Box: the red channel of the center pixel. (b) S_Red_Dot: a randomly selected red channel. (c) M_Red_Dot: 288 randomly selected red channels. (d) M_RGB_Dot: all three color channels of 96 randomly selected pixels ($s = 3 \times 96 = 288$). (e) M_Red_Eye: 288 red channels around the eyes of the animals. (f) M_RGB_Box: all colors of a centered 8×12 rectangle. (g) L_RGB_Box: all colors of a centered 101×101 rectangle. (h) X_RGB_Box: all dimensions. (i) FGSM. (j) PGD.

image x when they think it became just noticeably different from the original image $\mathbf{x}_0$. The participants had to hold $\mathbf{x}_0$ in memory, though they could also go all the way left back to see $\mathbf{x}_0$ again. We hosted the experiment using the NEXT platform [62, 136].

In a ± 1-perturbation trial, the perturbation direction $\mathbf{v}$ is one of the eight ± 1-perturbations. We allowed the participants to vary a within $\{0, 1, \ldots, 128\}$ to avoid value cropping. If a participant was not able to detect any change even after $a = 128$, then they were encouraged to "give up".

In an adversarial trial, the perturbation direction is $\mathbf{v}^{FGSM}$ or $\mathbf{v}^{PGD}$. We allowed the participants to increment a indefinitely, though no one went beyond $a = 80$(Figure 6.8).

The guard trials were designed to filter out participates who clicked through the experiment without performing the task. In a guard trial, we showed a novel fixed natural image (not panda, macaw or cat) for $a < 20$. Then for $a \geq 20$, a highly noisy version of that image

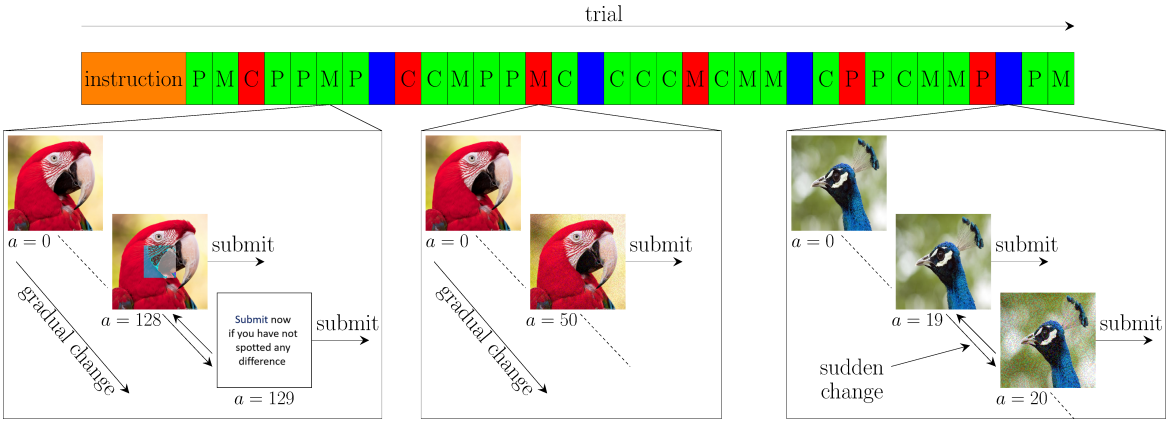


Fig. 6.7 Experiment procedure. The green, red and blue cells denote ± 1-perturbation, adversarial, and guard trials, respectively. The letters P, M and C denote the panda, macaw and cat $\mathbf{x}_0$, respectively.

is displayed. An attentive participant should readily notice this sudden change at $a = 20$ and submit it. We disregarded guard trials in our analysis.

Participants and data inclusion criterion: We enrolled 68 participants using Amazon Mechanical Turk [20] master workers. A master worker is a person who has consistently displayed a high degree of success in performing a wide range of tasks. All participants used a desktop, laptop or a tablet device; none used a mobile device where the screen would be too small. On average the participants took 33 minutes to finish the experiment. Each participant was paid \$5. As mentioned before, we use guard trials to identify inattentive participants. While the change happens at exactly $a = 20$ in a guard trial, our data indicates a natural spread in participant submissions around 20 with sharp decays. We speculate that the spread was due to keyboard / mouse auto repeat. We set a range for an acceptable guard trial if a participant submitted $a \in \{18, 19, 20, 21, 22\}$. A participant is deemed inattentive if any one of the four guard trials was outside the acceptable range. Only $n = 42$ out of 68 participants survived this stringent inclusion condition. All our analyses below are on these 42 participants.

To summarize the data: on each combination of natural image $\mathbf{x}_0$ and perturbation direction $\mathbf{v}$, the n participants gave us their individual perturbation scale $a^{(1)}, \ldots, a^{(n)}$. That is, the image $\mathbf{x} = \Pi(\mathbf{x}_0 + a^{(j)}\mathbf{v})$ is the one participant j thinks has just-noticeable-difference to $\mathbf{x}_0$ (human JND images). We present box plots of the data in Figure 6.8. The perturbation directions $\mathbf{v}$ are indicated on the x-axis. The box plots (left y-axis) show the median, quartiles, and outliers of the participants' perturbation scale a.

Because our participants can sometimes choose to "give up" if they did not notice a change, we have *right censored data* on a. All we know from a give-up trial is that

$a \geq 129$, but not what larger a value will cause the participant to noticed a difference. In Figure 6.8 the blue bars (right y-axis) show the number of participants who chose to "give up". Not surprisingly, many participants failed to notice a difference along the S_Red_Box and S_Red_Dot perturbation directions. Because of the presence of censored data, in later sections we often employ the Kolmogorov-Smirnov test which is a nonparametric test of distribution that can incorporate the censored data. There are 9 tests in total. To achieve a paper-wide significance level of e.g. $\alpha = 0.01$, we perform Bonferroni correction for multiple tests leading to individual test level $\alpha/9$.

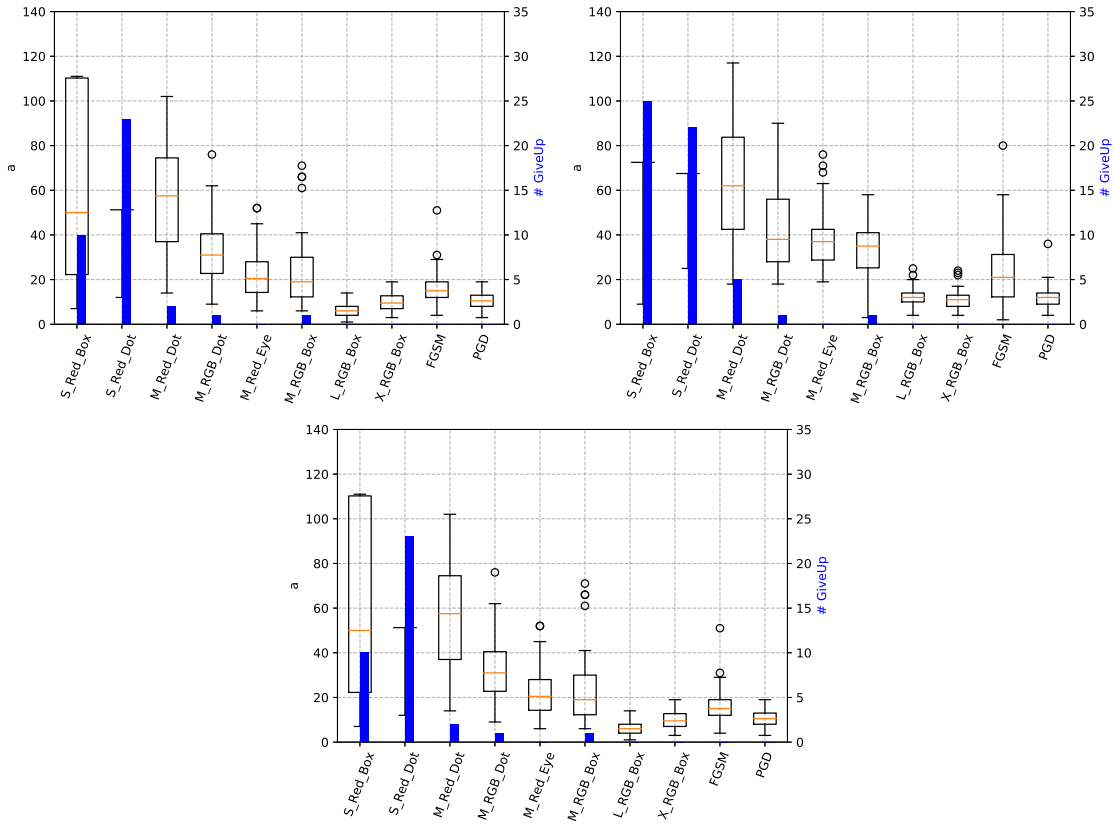


Fig. 6.8 Summary of data for $\mathbf{x}_0 =$ panda, macaw, cat, respectively

6.3 Pixel p-norms and other measures do not match human perception

6.3.1 Humans probably do not use pixel 1-norm, 2-norm, or ∞-norm

Let us start by assuming humans use pixel 1-norm. Implication 1 suggests the following procedure: for all center images $\mathbf{x}_0$, for all perturbation directions $\mathbf{v}_1, \mathbf{v}_2$, perturb $\mathbf{x}_0$ along these two directions separately until the images each become just noticeable to the ideal observer. Denote $t_1 \coloneqq \|\mathbf{x}_1 - \mathbf{x}_0\|_1$ and $t_2 \coloneqq \|\mathbf{x}_2 - \mathbf{x}_0\|_1$ on the two resulting images $\mathbf{x}_1, \mathbf{x}_2$. Then we have $t_1 = t_2$. Conversely, if the equality does not hold on even one triple ($\mathbf{x}_0$, $\mathbf{v}_1, \mathbf{v}_2$), then implication 1 with pixel 1-norm, and consequently the central hypothesis with pixel 1-norm, will be refuted.

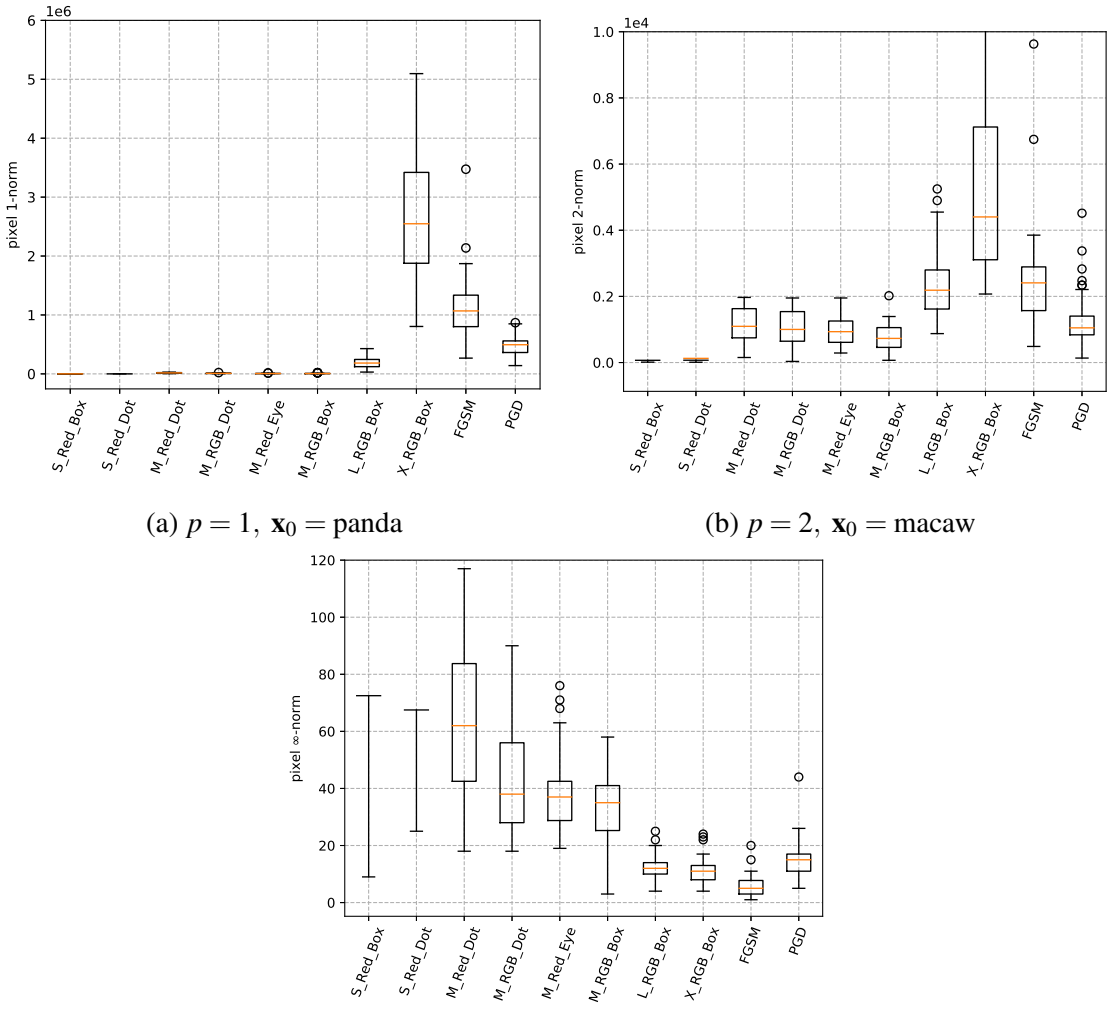
Of course, we do not have the ideal observer. Instead, we have n participants from the population. Starting from $\mathbf{x}_0$ along perturbation direction $\mathbf{v}_1$, the jth participant identifies their own just-noticeably-different image $\mathbf{x}_1^{(j)}$. Under pixel 1-norm this produces a number $t_{1j} \coloneqq \|\mathbf{x}_1^{(j)} - \mathbf{x}_0\|_1$. The numbers from all participants form a sample $\{t_{11}, \ldots, t_{1n}\}$ (there can be identical values). Similarly, denote the sample for direction $\mathbf{v}_2$ by $\{t_{21}, \ldots, t_{2n}\}$. Figure 6.9a shows a box plot for $\mathbf{x}_0 =$ panda. If implication 1 with pixel 1-norm were true, the medians (orange lines) would be at about the same height within the plot. Qualitatively this is not the case: the median for $\mathbf{v}_1 =$ FGSM is $t = 1068581$ but the median for $\mathbf{v}_2 =$ M-RGB-Dot is merely $t = 8928$. We perform a statistical test.

Hypothesis test 1 *The null hypothesis H_0 is: $\|\mathbf{x}_1^{(j)} - \mathbf{x}_0\|_1$ and $\|\mathbf{x}_2^{(j)} - \mathbf{x}_0\|_1$ have the same distribution, where $\mathbf{x}_0 =$ panda, $\mathbf{v}_1 =$ FGSM and $\mathbf{v}_2 =$ M-RGB-Dot. A two-sample Kolmogorov-Smirnov (KS) test on our data ($n = 42$) yields a p-value 6.4×10^{-19}, rejecting H_0.*

Exactly the same reasoning applies if we assume humans use pixel 2- or ∞-norm. Figure 6.9b shows pixel 2-norm on $\mathbf{x}_0 =$ macaw, where $\mathbf{v}_1 =$ PGD has median $t \approx 1049$ but $\mathbf{v}_2 =$ X-RGB-Box has median $t \approx 4402$; Figure 6.9c shows pixel ∞-norm on $\mathbf{x}_0 =$ cat, where $\mathbf{v}_1 =$ M-RGB-Box has median $t = 35$ but $\mathbf{v}_2 =$ L-RGB-Box has median $t = 12$. The full plots are in appendix Figure 6.13.

Hypothesis test 2 *H_0: $\|\mathbf{x}_1^{(j)} - \mathbf{x}_0\|_2$ and $\|\mathbf{x}_2^{(j)} - \mathbf{x}_0\|_2$ have the same distribution, where $\mathbf{x}_0 =$ macaw, $\mathbf{v}_1 =$ PGD and $\mathbf{v}_2 =$ X-RGB-Box. KS test yields p-value 2.6×10^{-16}, rejecting H_0.*

Hypothesis test 3 *H_0: $\|\mathbf{x}_1^{(j)} - \mathbf{x}_0\|_\infty$ and $\|\mathbf{x}_2^{(j)} - \mathbf{x}_0\|_\infty$ have the same distribution, where $\mathbf{x}_0 =$ cat, $\mathbf{v}_1 =$ M-RGB-Box and $\mathbf{v}_2 =$ L-RGB-Box. KS test yields p-value 1.1×10^{-14}, rejecting H_0.*



(a) $p = 1$, $\mathbf{x}_0 =$ panda

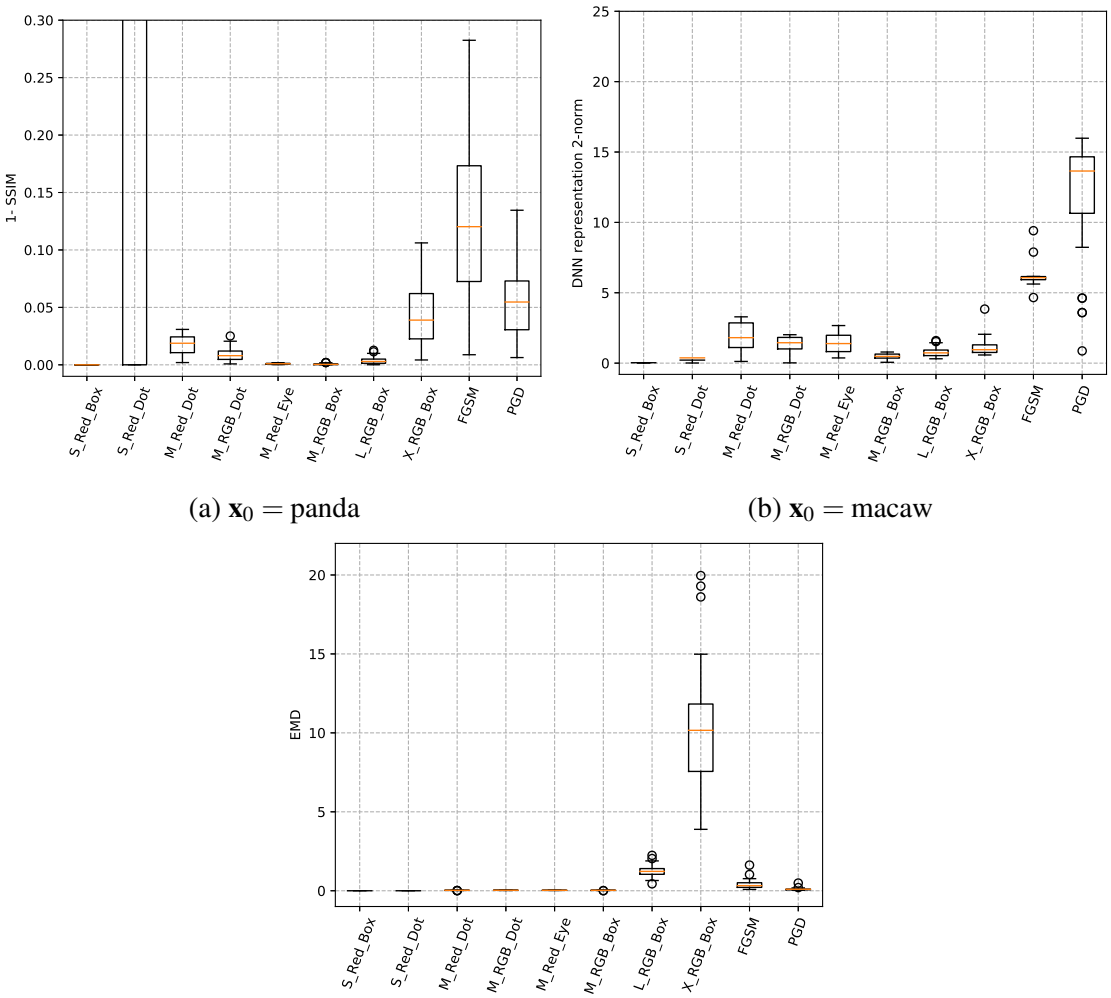
(b) $p = 2$, $\mathbf{x}_0 =$ macaw

(c) $p = \infty$, $\mathbf{x}_0 =$ cat

Fig. 6.9 Participant JND $\mathbf{x}$'s pixel p-norm $\|\mathbf{x} - \mathbf{x}_0\|_p$. If the central hypothesis were true, one expects a plot to have similar medians (orange lines).

6.3.2 Humans probably do not use any pixel p-norm

But what if humans use some other pixel p-norm in $[0, \infty)$? Implication 1 requires a specific p to test, which is not convenient. Instead, we now test implication 2 whose failure can refute any $p \in [0, \infty]$. We take $\mathbf{x}_0 =$ cat and look at the two perturbation directions $\mathbf{v}_1 =$ M-Red-Dot and $\mathbf{v}_2 =$ M-Red-Eye. These two perturbations have the same sparsity $s = 288$. Therefore, implication 2 predicts that the scales a_1, a_2 to reach just-noticeable-difference should be the same. However, the perturbation directions differ in their "shape of support": M-Red-Dot changes random pixels, while M-Red-Eye changes pixels of the eye region which presumably humans pay attention to and thus detect earlier. On perturbation direction $\mathbf{v}_1$,



(a) $\mathbf{x}_0 =$ panda

(b) $\mathbf{x}_0 =$ macaw

(c) $\mathbf{x}_0 =$ cat

Fig. 6.10 Box plots of 1 - SSIM, DNN representation and EMD respectively on human JND images.

our n participants produced scales $\{a_1^{(1)}, \ldots, a_1^{(n)}\}$; similarly, for the other direction $\mathbf{v}_2$, they produced $\{a_2^{(1)}, \ldots, a_2^{(n)}\}$. See Figure 6.8(bottom) for the human behaviors: the median scale is 62 and 37, respectively, as we suspected.

Hypothesis test 4 *H_0: Human JND $a_1^{(j)}$ and $a_2^{(j)}$ have the same distribution, where $\mathbf{x}_0 =$ cat, $\mathbf{v}_1 =$ M-Red-Dot and $\mathbf{v}_2 =$ M-Red-Eye. KS test yields p-value 9.5×10^{-6}, rejecting H_0.*

We also take $\mathbf{x}_0 =$ panda and look at the two perturbation directions $\mathbf{v}_1 =$ M-Red-Dot and $\mathbf{v}_2 =$ M-RGB-Dot. The directions again have the same sparsity; this time they also share the same "shape of support": the nonzero elements of $\mathbf{v}_1, \mathbf{v}_2$ are both randomly scattered over pixels. The difference is that $\mathbf{v}_1$ changes only the red color channel on 288 random pixels,

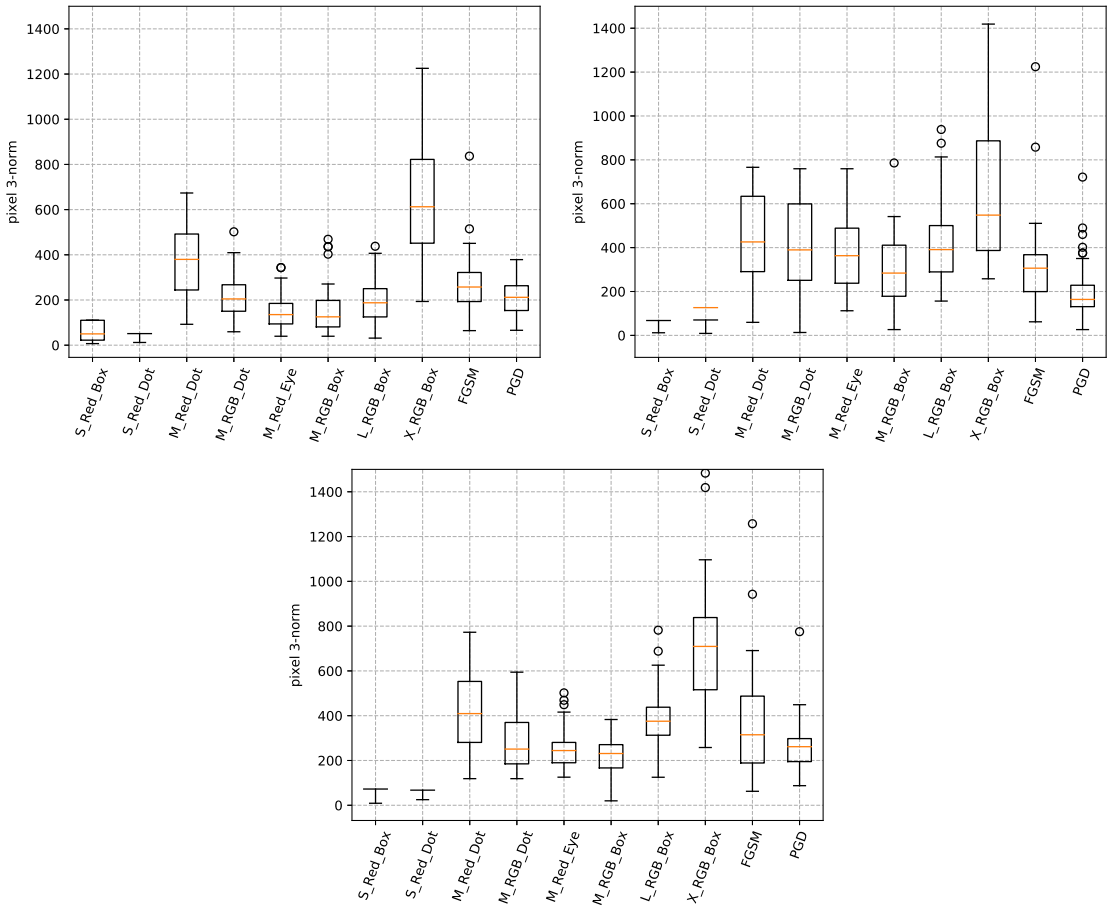


Fig. 6.11 Participants JND pixel 3-norm $\|\mathbf{x}-\mathbf{x}_0\|_3$ for panda, macaw, and cat, respectively.

while $\mathbf{v}_2$ changes all three channels but only on $288/3 = 96$ random pixels. Implication 2 again predicts that the humans should reach just-noticeable-difference at the same scale. But Figure 6.8(top left) suggest that humans are more sensitive to simultaneous changes to all RGB channels: scales a_1, a_2 have median 57.5 and 31, respectively.

Hypothesis test 5 *The null hypothesis H_0 is: The sample $\{a_1^{(1)},\ldots,a_1^{(n)}\}$ generated from $\mathbf{x}_0 =$ panda, $\mathbf{v}_1 =$ M-Red-Dot and the sample $\{a_2^{(1)},\ldots,a_2^{(n)}\}$ generated from $\mathbf{x}_0 =$ panda, $\mathbf{v}_2 =$ M-RGB-Dot come from the same distribution. A two-sample Kolmogorov-Smirnov test on our data ($n = 42$) yields a p-value 2.1×10^{-4}, rejecting H_0.*

These tests refute implication 2.

To further strengthen our case for pixel p-norms, we also test implication 3 which states that it is easier to notice changes if the perturbation $\mathbf{v}$ has larger support s. This is mostly true as seen in Figure 6.8: the median of a generally decreases as $\mathbf{v}$ support size increases in the order of S, M, L, X. However, there is a curious inversion on $\mathbf{x}_0 =$ panda, $\mathbf{v}_1 =$ L-RGB-Box

vs $\mathbf{v}_2 =$ X-RGB-Box: Implication 3 predicts that $a_1 > a_2$, but human behaviors have mean 6.5 and 9.9 (and median 6 and 9.5), respectively: the other way around. The human data for these perturbations are not censored; Figure 6.8 also suggests they are close to normal in distribution. We therefore perform a one-tailed two-sample t-test with unequal variances.

Hypothesis test 6 *The null hypothesis H_0 is: Human JND scales generated from $\mathbf{x}_0 =$ panda, $\mathbf{v}_1 =$ L-RGB-Box has equal mean as those generated from $\mathbf{x}_0 =$ panda, $\mathbf{v}_2 =$ X-RGB-Box. The left-tailed alternative hypothesis H_a is: the former has a smaller mean. A one-tailed two-sample t-test with unequal variances on our data $(n = 42)$ yields a p-value of 1.8×10^{-5}, rejecting H_0 and retaining H_1.*

This test suggests that the inversion is statistically significant, thus refuting implication 3. We speculate that the inversion is due to the black-and-white panda making the L box boundary more prominent, see Figure 6.6(g).

Taken together, these results indicate that pixel p-norms are not a good fit for human behaviors regardless of p. There are probably other perceptual attributes that are important to humans which are unaccounted for by pixel p-norms.

6.3.3 EMD, 1- SSIM and DNN Representation also do not match human perception

We now test the remaining measures. Figure 6.10c shows the box plots of human JND images $EMD(\mathbf{x}, \mathbf{x}_0)$ along different perturbation directions for $\mathbf{x}_0 =$ cat (the full plots are in appendix Figure 6.14). It is immediately clear that on perturbation direction X-RGB-Box humans need to move a lot more earth as measured by EMD before they perceive the image difference. These should not happen: ideally human JND should occur at the same $\rho(\mathbf{x}, \mathbf{x}_0)$ value. The following test implies EMD probably should not be used.

Hypothesis test 7 *H_0: Human JND images' $EMD(\mathbf{x}, \mathbf{x}_0)$ on directions $\mathbf{v}_1 =$ M-RGB-Box, $\mathbf{v}_2 =$ X-RGB-Box for $\mathbf{x}_0 =$ cat have the same distribution. KS test yields p-value 6.4×10^{-19}, rejecting H_0.*

Figure 6.10a shows the box plots of $1 - SSIM(\mathbf{x}, \mathbf{x}_0)$ of our participant data for $\mathbf{x}_0 =$ panda (The full plot is in appendix Figure 6.15). The following test implies 1 - SSIM probably should not be used to define adversarial attack detectability.

Hypothesis test 8 *H_0: Human JND $1 - SSIM(\mathbf{x}, \mathbf{x}_0)$ on directions $\mathbf{v}_1 =$ X-RGB-Box, $\mathbf{v}_2 =$ FGSM for $\mathbf{x}_0 =$ panda have the same distribution. KS test yields p-value 1.1×10^{-9}, rejecting H_0.*

Figure 6.10b shows the box plots of human JND images' DNN 2-norm along different perturbation directions for $\mathbf{x}_0 =$ macaw. The full plot for DNN $p = 1, 2, \infty$ norms and all animals is in appendix Figure 6.16. Interestingly, the human JND images along the adversarial perturbation directions (FGSM and PGD) have much larger DNN p-norm than the ± 1 perturbation directions. As an example, $\mathbf{x}_0 =$ macaw, $\mathbf{v}_1 =$ M-Red-Dot human JND images have median DNN 2-norm 1.8, while $\mathbf{v}_2 =$ PGD human JND images have median 13.6. The following test implies that 2-norm on DNN representation probably should not be used to define adversarial attack detectability.

Hypothesis test 9 *H_0: Human JND images' DNN 2-norm along $\mathbf{v}_1 =$ M-Red-Dot and $\mathbf{v}_2 =$ PGD for $\mathbf{x}_0 =$ macaw have the same distribution. KS test yields p-value 2.1×10^{-12}, rejecting H_0.*

6.4 Ranking the different measures

We emphasize that our human experiments do *not* support pixel p-norm, EMD, 1 - SSIM, or DNN representation as a *good fit* for human perception. Hence, using any of them will lead to type I and II errors, examples of which were showed in Figures 6.2 and 6.3 respectively. Nonetheless, some of them may be *useful* as computational approximations to human perception. As such, in this section we rank the measures and identify the measure which offers the best approximation. While none of the measures exactly satisfies $\forall \mathbf{x}_1, \mathbf{x}_2 \in J(\mathbf{x}_0), \rho(\mathbf{x}_1, \mathbf{x}_0) = \rho(\mathbf{x}_2, \mathbf{x}_0)$, the equality inspires the following idea: the best measure should minimize the standard deviation of $\rho(\mathbf{x}, \mathbf{x}_0)$ over all human JND images $\mathbf{x} \in J(\mathbf{x}_0)$. This is because $\rho(\mathbf{x}, \mathbf{x}_0)$ would have been a constant if the equality were true. However, different measures have vastly different scales (e.g. for p-norms alone, the all-1 vector in R^d has 1-norm d, 2-norm $\sqrt{d}$, and ∞-norm 1), making a direct comparison difficult. Instead, we normalize by the center image $\mathbf{x}_0$ in order to find the best approximation: $\min_\rho \text{std}\left(\frac{\rho(\mathbf{x}, \mathbf{x}_0)}{\rho(\mathbf{x}_0, \mathbf{0})}\right)$ where $\mathbf{0}$ is the zero vector. The standard deviation is taken over all our human experiment data for a particular center image $\mathbf{x}_0$, pooling all participants and all perturbation directions together, excluding "give ups". Figure 6.4 shows $\text{std}\left(\frac{\rho(\mathbf{x}, \mathbf{x}_0)}{\rho(\mathbf{x}_0, \mathbf{0})}\right)$ of different measures. For pixel p-norms this is presented as a function of p; EMD and 1 - SSIM are constant lines; and DNN has values larger than 0.12 for all p and thus not shown Figure 6.4. We show the results of DNN representations in Figure 6.12

Interestingly, by this criterion the pixel 3-norm is the best approximation of human JND judgment among the tested measures. We plot $\|\mathbf{x} - \mathbf{x}_0\|_3$ of the human JND $\mathbf{x}$'s in Figure 6.11. Compared to pixel 1, 2, and ∞ norms in Figure 6.9, EMD, 1 - SSIM, and DNN p-norm in Figure 6.10, and their full plots in the appendix, the median of pixel 3-norm (orange lines)

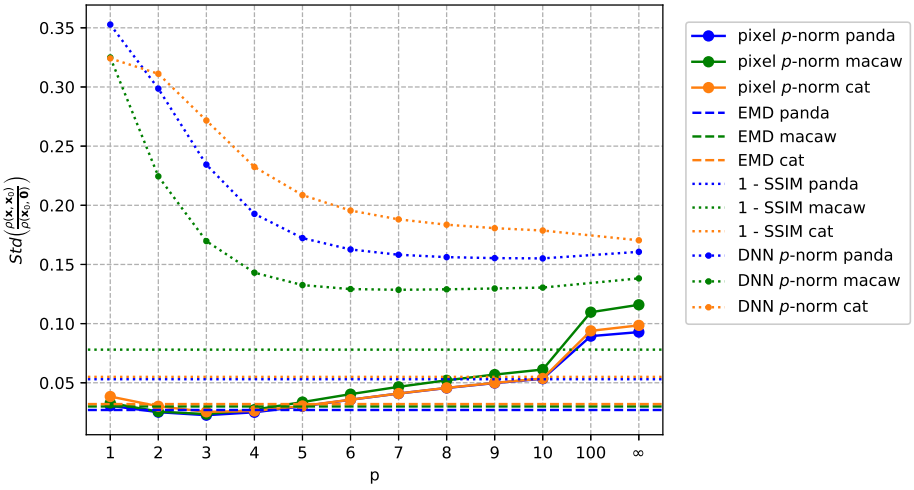


Fig. 6.12 Variability of fit to human data including DNN representations, lower is better

are closer to having the same height. This qualitatively supports pixel 3-norm as a better approximation than the other measures. The rank of the rest of the measures are: pixel 2-norm, EMD, pixel 1-norm, 1 - SSIM, pixel ∞-norm and DNN representation respectively.

6.5 Conclusion

Our behavioral experiment suggests that pixel p-norms, EMD, 1 - SSIM, and DNN representation p-norms do not match how humans judge just-noticeably-different images. However, we rank the different measures and found pixel 3-norm to be the closest approximation. Future research is needed to identify better measures of cognitive response to image distortion, and to generalize our work to other domains such as audio and text.

6.6 Contribution

In this project, I contributed in the human experiment design. Furthermore, I was responsible for running the human experiment. I also helped analyzing the data and proposing the hypothesis tests. Robert Nowak, Electrical and Computer Engineering, University of Wisconsin-Madison helped us in designing the human experiment presented in this work.

6.7 Supplemental materials

6.7.1 More plots

We present further plots in this section.

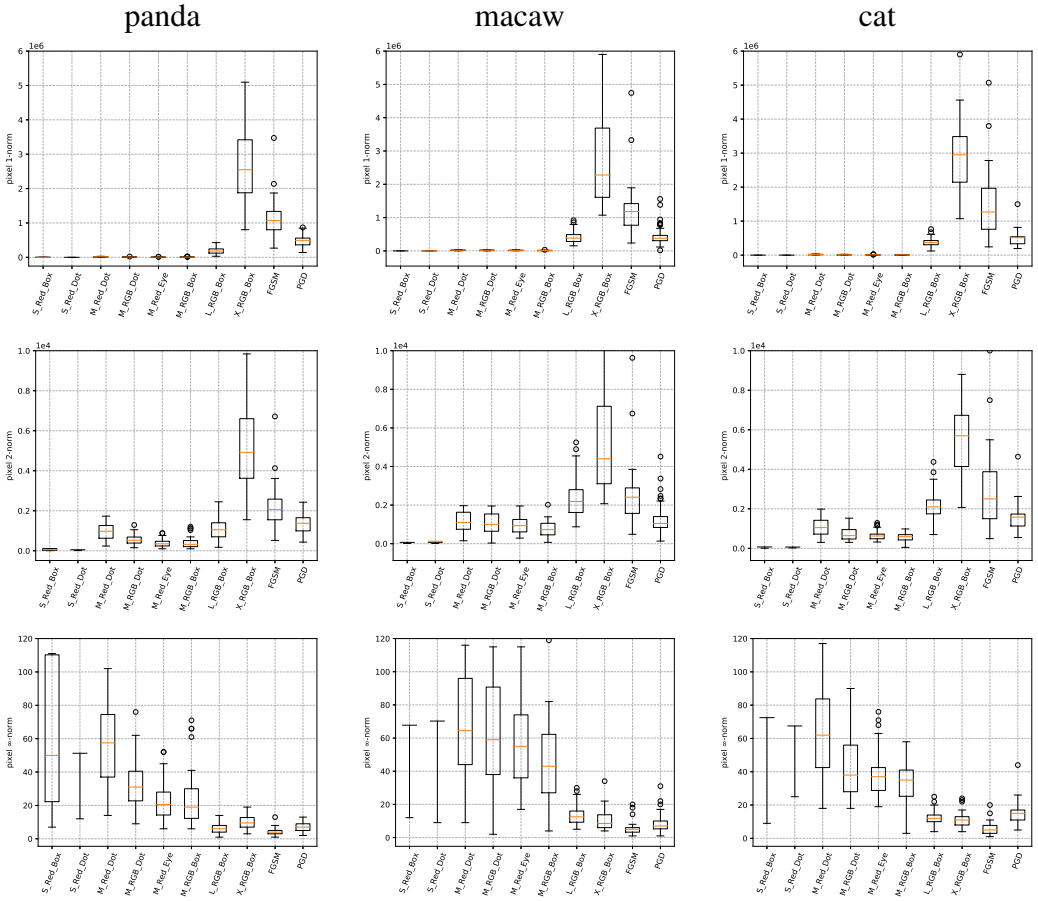
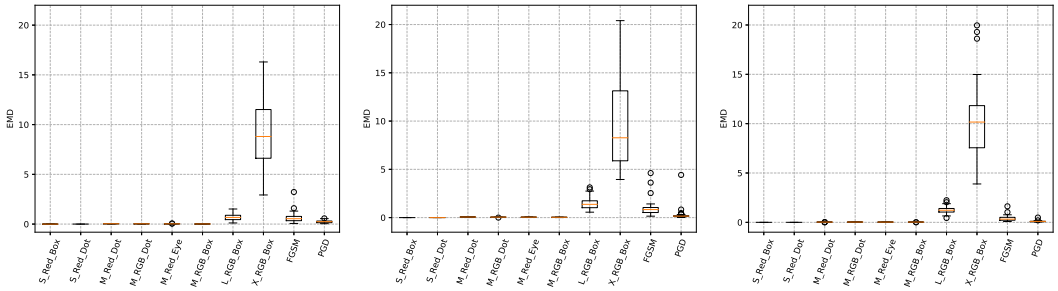


Fig. 6.13 Participant JND $\mathbf{x}$'s pixel p-norm $\|\mathbf{x}-\mathbf{x}_0\|_p$ for $p = 1$ (top row), 2 (middle row), ∞ (bottom row). Within a plot, each vertical box is for a perturbation direction $\mathbf{v}$. The box plot depicts the median, quartiles, and outliers. If the central hypothesis were true, one expects a plot to have similar medians (orange lines).

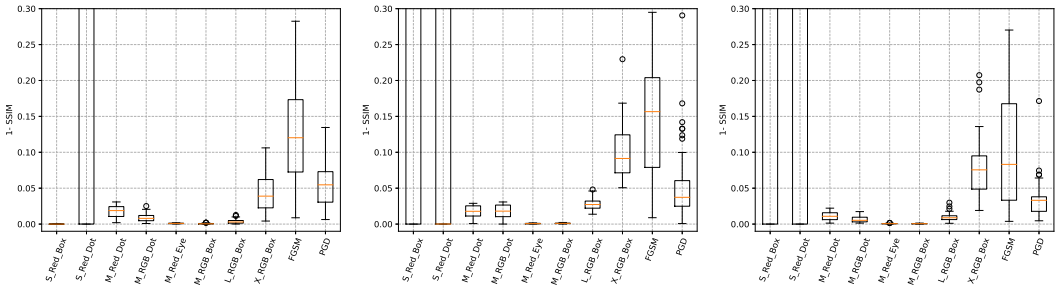
6.7.2 Amazon Mechanical Turk instructions

For reference, screenshots of the instructions displayed to participants are included in this appendix.



(a) panda (b) macaw (c) cat

Fig. 6.14 Box plots of Earth Mover's Distance on human JND images. Recall for each natural image $\mathbf{x}_0$ and each perturbation direction $\mathbf{v}$, our n participants decided which image $\mathbf{x}^{(j)} = \Pi(\mathbf{x}_0 + a^{(j)}\mathbf{v})$ is JND to them, for $j = 1 \dots n$. We compute $EMD(\mathbf{x}^{(1)}, \mathbf{x}_0), \dots EMD(\mathbf{x}^{(n)}, \mathbf{x}_0)$ and show them as a box plot. Doing so for all our perturbation directions $\mathbf{v}$ and all natural images $\mathbf{x}_0$ produces this figure.



(a) panda (b) macaw (c) cat

Fig. 6.15 Box plots of 1 - SSIM on human JND images.

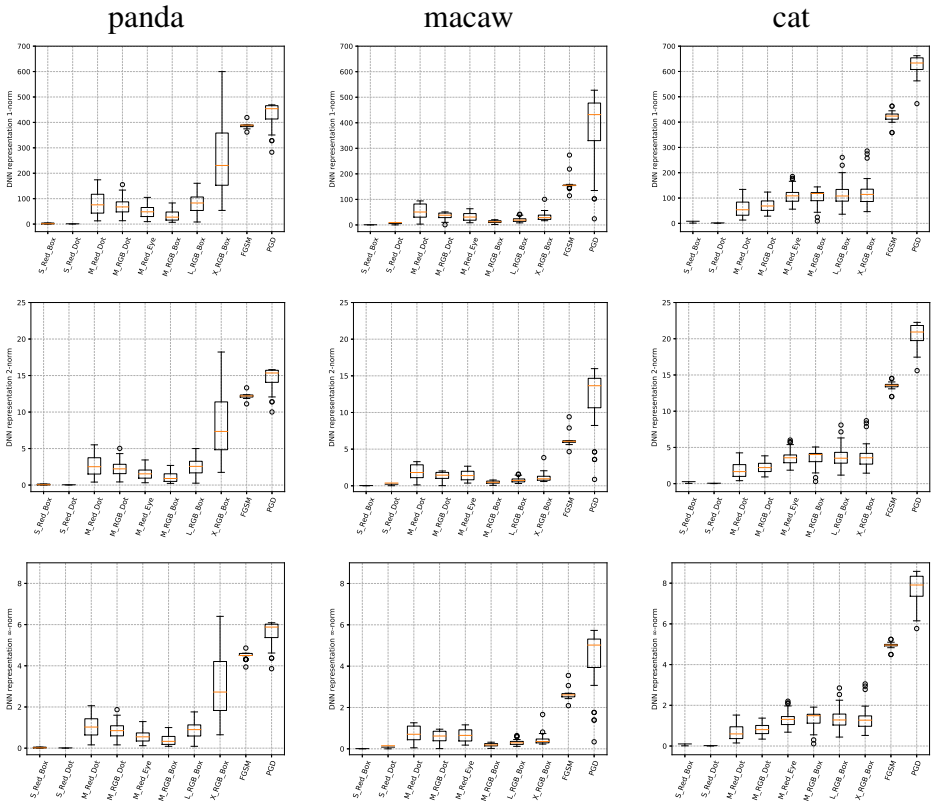


Fig. 6.16 Box plots of DNN $\|\xi(\mathbf{x}) - \xi(\mathbf{x}_0)\|_p$ on human JND images. rows: $p = 1, 2, \infty$, respectively.

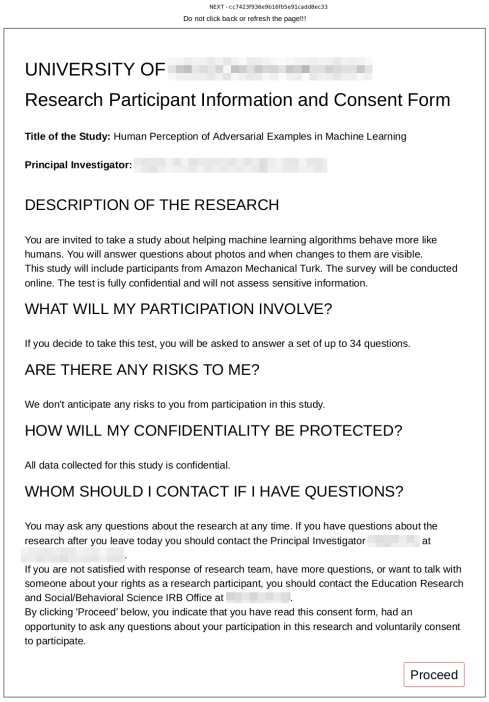


Fig. 6.17 Instruction Page 1

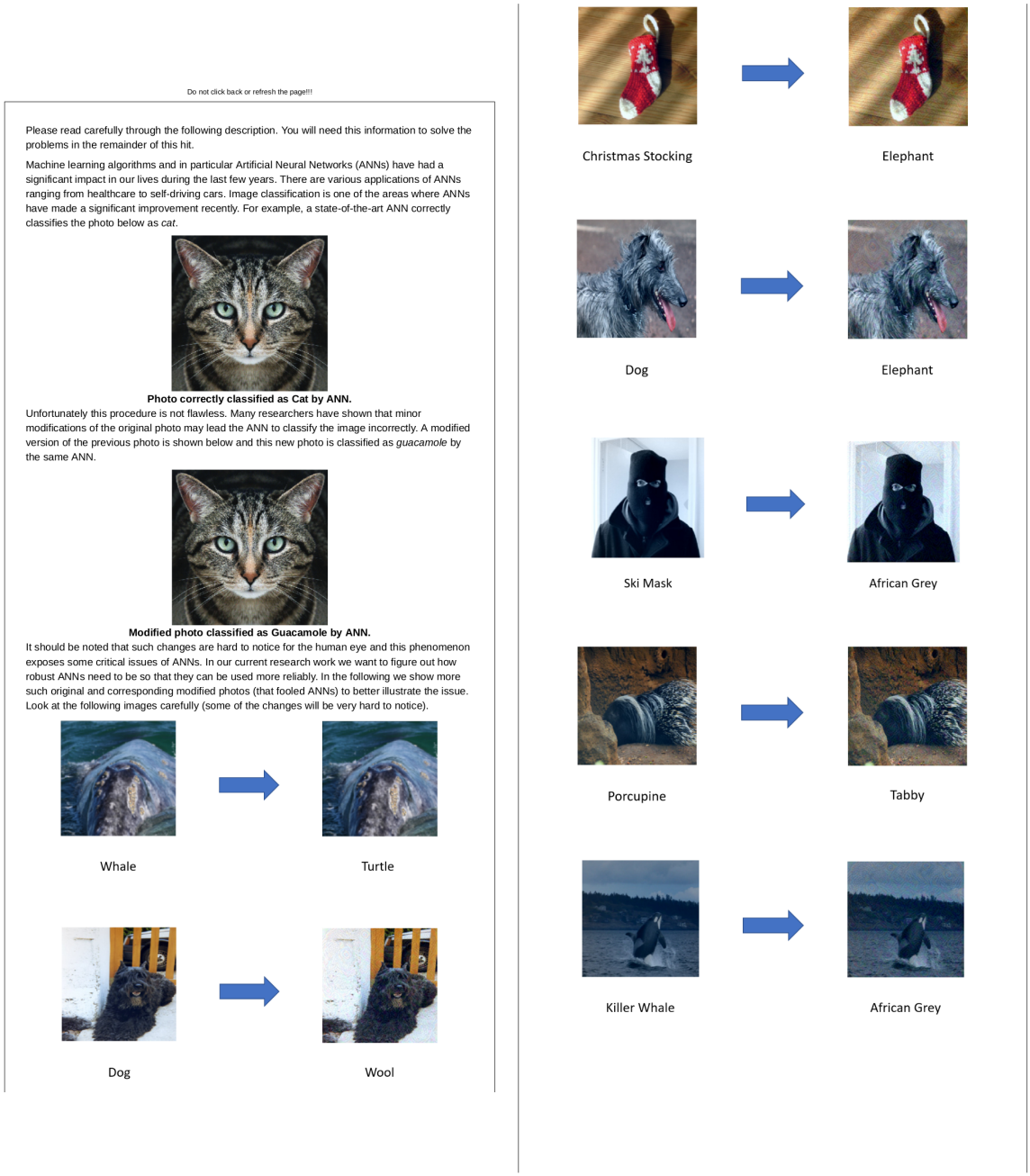


Fig. 6.18 Instruction Page 2

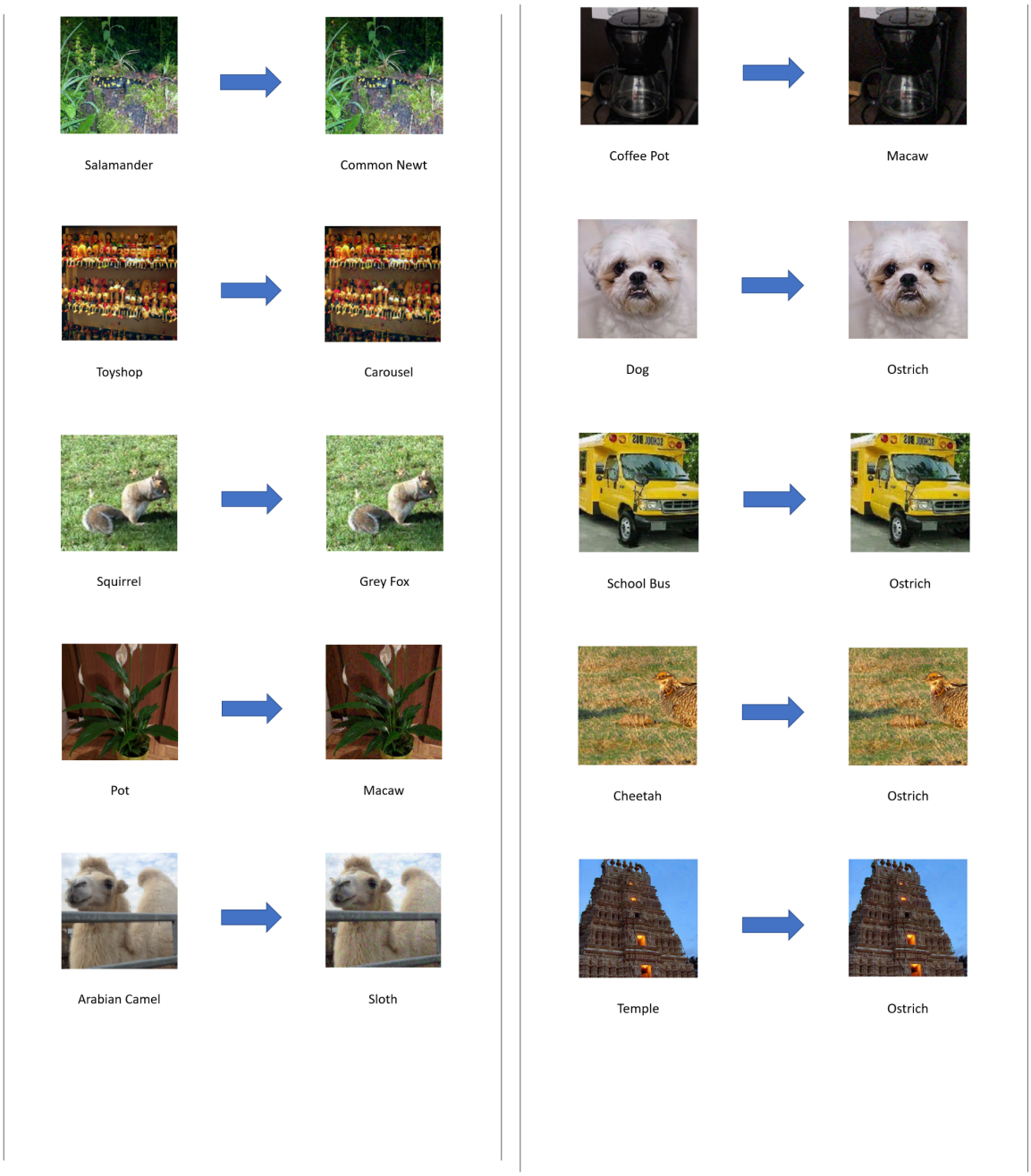


Fig. 6.19 Instruction Page 2 (*cont'd*)

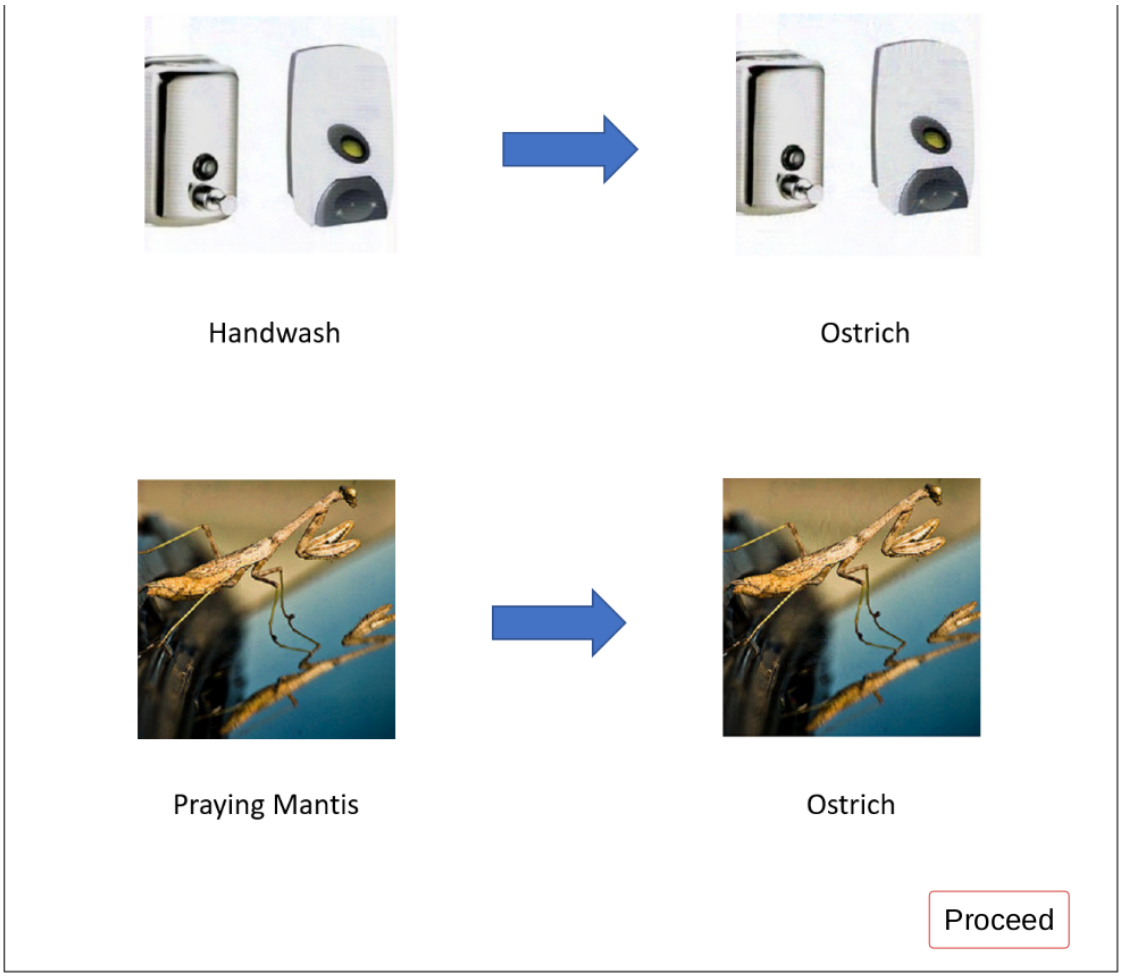
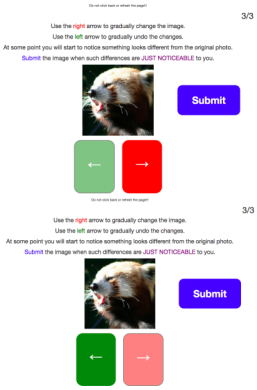


Fig. 6.20 Instruction Page 2 (*cont'd*)

NEXT - cc7423f936e9b16fb5e91cadd8ec33

Do not click back or refresh the page!!!

In this hit you will answer questions regarding when changes to a photo becomes just noticeable. In each question you will be presented with an unmodified photo. Using the **right arrow key** on your keyboard you will be able to gradually change the photo. The changes will apply to **pixel values** in the photo. **No rotation, cropping or other transformations will be applied. You should stop as soon as you notice changes to the photo**. Then press the **enter key** (or click **Submit**) to submit this modified photo and move to the next one. If you realize that you have pressed the right arrow key too many times (i.e., the changes were visible earlier), then you can press the **left arrow key** to gradually undo the changes. You can click the left/right arrows instead of pressing left/right arrow key to achieve the same effect. You will also receive visual cues (check the following figures) indicating that the system is detecting when you press the right and left arrow keys.



Left and right arrow key press indicators.

There are **34 questions** in total. Some unmodified photos are repeated multiple times. But changes applied to these photos will be different each time. The hit will take approximately **30 minutes** to complete. Note that there are a few questions where the point of change is obvious and others where the changes may be more subtle. The payment for successfully completing the hit will depend on how you perform on all questions. Once you are ready click proceed.

Proceed

Fig. 6.21 Instruction Page 3

References

[1] (1997). Perceptual learning. *San Diego, CA: Academic Press*.

[2] Ainsworth, S. (2006). DeFT: A conceptual framework for considering learning with multiple representations. *Learning and instruction*, 16(3):183–198.

[3] Ainsworth, S. (2008). The educational value of multiple-representations when learning complex scientific concepts. *Visualization: Theory and practice in science education*, pages 191–208.

[4] Alfeld, S., Zhu, X., and Barford, P. (2016). Data poisoning attacks against autoregressive models. In *AAAI*, pages 1452–1458.

[5] Alfeld, S., Zhu, X., and Barford, P. (2017). Explicit defense actions against test-set attacks. In *AAAI*, pages 1274–1280.

[6] Athalye, A., Carlini, N., and Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*.

[7] Attwell, D. and Laughlin, S. B. (2001). An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism*, 21(10):1133–1145.

[8] Baddeley, A. (1992). Working memory. *Science*, 255(5044):556–559.

[9] Baluja, S. and Fischer, I. (2017). Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*.

[10] Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. (2010). The security of machine learning. *Machine Learning*, 81(2):121–148.

[11] Barreno, M., Nelson, B., Sears, R., Joseph, A. D., and Tygar, J. D. (2006). Can Machine Learning Be Secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*.

[12] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. *Proceedings of the 26th International Conference On Machine Learning, ICML 2009*, pages 41–48.

[13] Bhattacharya, T., Dey, N., and Chaudhuri, S. (2012). A session based multiple image hiding technique using dwt and dct. *arXiv preprint arXiv:1208.0950*.

[14] Bodemer, D., Ploetzner, R., Feuerlein, I., and Spada, H. (2004). The active integration of information during learning with dynamic and interactive visualisations. *Learning and Instruction*, 14(3):325–341.

[15] Botvinick, M. and Plaut, D. C. (2004). Doing without schema hierarchies: a recurrent connectionist approach to normal and impaired routine sequential action. *Psychological review*, 111(2):395.

[16] Braden, R. A. (1997). Handbook of research for educational communications and technology.

[17] Brakerski, Z. (2012). Fully homomorphic encryption without modulus switching from classical gapsvp. In *Advances in cryptology–crypto 2012*, pages 868–886. Springer.

[18] Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (2014). (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):13.

[19] Brysbaert, M. and New, B. (2009). Moving beyond Kučera and Francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for American English. *Behavior Research Methods*, 41(4):977–990.

[20] Buhrmester, M., Kwang, T., and Gosling, S. D. (2011). Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5.

[21] Bybee, J. and McClelland, J. L. (2005). Alternatives to the combinatorial paradigm of linguistic theory based on domain general principles of human cognition. *Linguistic Review*, 22(2-4):381–410.

[22] Cachin, C. (1998). An information-theoretic model for steganography. In *Information Hiding*, pages 306–318. Springer.

[23] Cakmak, M. and Thomaz, A. L. (2014). Eliciting good teaching from humans for machine learners. *Artificial Intelligence*, 217:198–215.

[24] Carlini, N. and Wagner, D. (2017a). Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM.

[25] Carlini, N. and Wagner, D. (2017b). Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE.

[26] Chandramouli, R. (2002). A mathematical approach to steganalysis. In *Proc. SPIE*, volume 4675, pages 14–25.

[27] Chase, W. G. and Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, 4(1):55–81.

[28] Cox, C. R., Borkenhagen, M. C., and Seidenberg, M. S. (2019). Efficiency of learning in experience-limited domains : Generalization beyond the WUG test. In Goel, A., Seifert, C., and Freksa, C., editors, *CogSci 2019*, pages 1566–1571.

[29] Cox, I. J., Kalker, T., Pakura, G., and Scheel, M. (2005). Information transmission and steganography. In *IWDW*, pages 15–29. Springer.

[30] Daemen, J. and Rijmen, V. (2013). *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media.

[31] Dalvi, N., Domingos, P., Sanghai, S., Verma, D., et al. (2004). Adversarial Classification. In *ACM SIGKDD*.

[32] Davies, M. (2010). The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and Linguistic Computing*, 25(4):447–464.

[33] Demuth, H. B., Beale, M. H., De Jess, O., and Hagan, M. T. (2014). *Neural network design*. Martin Hagan.

[34] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee.

[35] Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*.

[36] Eilam, B. (2012). *Teaching, learning, and visual literacy: The dual role of visual representation*. Cambridge University Press.

[37] Ekdahl, P. and Johansson, T. (2002). A new version of the stream cipher snow. In *International Workshop on Selected Areas in Cryptography*, pages 47–61. Springer.

[38] ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472.

[39] Elman, J. L. (1993). Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99.

[40] Engstrom, L., Tsipras, D., Schmidt, L., and Madry, A. (2017). A rotation and a translation suffice: Fooling cnns with simple transformations. *arXiv preprint arXiv:1712.02779*.

[41] Fechner, G. T., Boring, E. G., Howes, D. H., and Adler, H. E. (1966). *Elements of Psychophysics. Translated by Helmut E. Adler. Edited by Davis H. Howes And Edwin G. Boring, With an Introd. by Edwin G. Boring*. Holt, Rinehart and Winston.

[42] Flaxman, A. D., Kalai, A. T., and McMahan, H. B. (2004). Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*.

[43] Flesch, T., Balaguer, J., Dekker, R., Nili, H., and Summerfield, C. (2018). Comparing continual task learning in minds and machines. *Proceedings of the National Academy of Sciences*, 115(44):E10313–E10322.

[44] Gentner, D. and Markman, A. B. (1997). Structure mapping in analogy and similarity. *American psychologist*, 52(1):45.

[45] Gentry, C., Sahai, A., and Waters, B. (2013). Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013*, pages 75–92. Springer.

[46] Gibson, B. R., Rogers, T. T., Kalish, C., and Zhu, X. (2015). What causes category-shifting in human semi-supervised learning? In *CogSci*.

[47] Gibson, E. J. (2000). Perceptual learning in development: Some basic concepts. *Ecological Psychology*, 12(4):295–302.

[48] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323.

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

[50] Goldstone, R. L. and Barsalou, L. W. (1998). Reuniting perception and conception. *Cognition*, 65(2):231–262.

[51] Goldstone, R. L., Schyns, P. G., and Medin, D. L. (1997). Learning to bridge between perception and cognition. *The psychology of learning and motivation*, 36:1–14.

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

[53] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773.

[54] Gupta, D. and Choubey, S. (2015). Discrete wavelet transform for image processing. *International Journal of Emerging Technology and Advanced Engineering*, 4(3):598–602.

[55] Harel, A. (2016). What is special about expertise? visual expertise reveals the interactive nature of real-world object recognition. *Neuropsychologia*, 83:88–99.

[56] Harm, M. W., McCandliss, B. D., and Seidenberg, M. S. (2003). Modeling the Successes and Failures of Interventions for Disabled Readers. *Scientific Studies of Reading*, 7(2):155–182.

[57] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778.

[58] Hemdan, E., El Fishawy, N., Attiya, G., and El-Samie, F. (2013). An efficient image watermarking approach based on wavelet fusion and singular value decomposition in wavelet domain. In *Proceeding of 3rd International Conference on Advanced Control Circuits And Systems (ACCS'013)(2013b) Google Scholar*.

[59] Hopper, N. J., Langford, J., and Von Ahn, L. (2002). Provably secure steganography. In *Annual International Cryptology Conference*, pages 77–92. Springer.

[60] Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., and Tygar, J. (2011). Adversarial Machine Learning. In *AISEC*.

[61] Itti, L. and Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews: Neuroscience*, 2(3):194.

[62] Jamieson, K. G., Jain, L., Fernandez, C., Glattard, N. J., and Nowak, R. (2015). Next: A system for real-world development, evaluation, and application of active learning. In *Advances in Neural Information Processing Systems*, pages 2656–2664.

[63] Jang, U., Wu, X., and Jha, S. (2017). Objective metrics and gradient descent algorithms for adversarial examples in machine learning. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 262–277. ACM.

[64] Jha, S. and Seshia, S. A. (2017). A theory of formal synthesis via inductive learning. *Acta Informatica*, 54(7):693–726.

[65] Johnson, N. F. and Jajodia, S. (1998). Exploring steganography: Seeing the unseen. *Computer*, 31(2).

[66] Karlik, B. and Olgac, A. V. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122.

[67] Katz, J., Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (1996). *Handbook of applied cryptography*. CRC press.

[68] Kellman, P. J. and Massey, C. M. (2013). Perceptual learning, cognition, and expertise. *The psychology of learning and motivation*, 58:117–165.

[69] Kerckhoffs, A. (1883a). "la cryptographie militaire (part i). volume 9, pages 5–38.

[70] Kerckhoffs, A. (1883b). "la cryptographie militaire (part ii). volume 9, pages 161–191.

[71] Kloft, M. and Laskov, P. (2010). Online anomaly detection under adversarial impact. In *AISTATS*, pages 405–412.

[72] Koedinger, K. R., Corbett, A. T., and Perfetti, C. (2012). The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science*, 36(5):757–798.

[73] Kornell, N. and Bjork, R. A. (2008). Learning concepts and categories: Is spacing the "enemy of induction"? *Psychological science*, 19(6):585–592.

[74] Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., Belongie, S., Gomes, V., Gupta, A., Sun, C., Chechik, G., Cai, D., Feng, Z., Narayanan, D., and Murphy, K. (2017). Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*.

[75] Krenn, R. (2004). Steganography and steganalysis.

[76] Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, 1:417–446.

[77] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.

[78] Kumaran, D., Hassabis, D., and McClelland, J. L. (2016). What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7):512–534.

[79] Kuperman, V., Stadthagen-Gonzalez, H., and Brysbaert, M. (2012). Age-of-acquisition ratings for 30,000 English words. *Behavior Research Methods*, 44(4):978–990.

[80] Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

[81] Lashley, K. S. (1951). *The problem of serial order in behavior*, volume 21. Bobbs-Merrill Oxford, United Kingdom.

[82] Laskov, P. and Kloft, M. (2009). A Framework for Quantitative Security Analysis of Machine Learning. In *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*.

[83] LeCun, Y., Cortes, C., and Burges, C. J. (1998). The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

[84] Liu, J. and Zhu, X. (2016). The teaching dimension of linear learners. *Journal of Machine Learning Research*, 17(162):1–25.

[85] Liu, W., Dai, B., Li, X., Rehg, J. M., and Song, L. (2017a). Towards black-box iterative machine teaching. *arXiv preprint arXiv:1710.07742*.

[86] Liu, W., Dai, B., Rehg, J. M., and Song, L. (2017b). Iterative machine teaching. *arXiv preprint arXiv:1705.10470*.

[87] López-Alt, A., Tromer, E., and Vaikuntanathan, V. (2012). On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1219–1234. ACM.

[88] Lowd, D. and Meek, C. (2005). Adversarial learning. In *ACM SIGKDD*, pages 641–647. ACM.

[89] Ma, Y., Nowak, R., Rigollet, P., Zhang, X., and Zhu, X. (2018). Teacher improves learning by selecting a training subset. *arXiv preprint arXiv:1802.08946*.

[90] Ma, Y., Zhang, X., Sun, W., and Zhu, J. (2019a). Policy poisoning in batch reinforcement learning and control. In *Advances in Neural Information Processing Systems*, pages 14543–14553.

[91] Ma, Y., Zhu, X., and Hsu, J. (2019b). Data poisoning against differentially-private learners: Attacks and defenses. *arXiv preprint arXiv:1903.09860*.

[92] MacWhinney, B. (2000). *The CHILDES Project: tools for analyzing talk.* Lawrence Erlbaum Associates, Mahwah, NJ, third edition.

[93] 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*.

[94] Marvel, L. M., Boncelet, C. G., and Retter, C. T. (1999). Spread spectrum image steganography. *IEEE Transactions on image processing*, 8(8):1075–1083.

[95] Mayer, R. E. (2009). *Cognitive theory of multimedia learning*. The cambridge handbook of multimedia learning (2nd ed., pp. 31-48). New York, NY: Cambridge University Press.

[96] McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

[97] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[98] Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. (2017). Universal adversarial perturbations. *arXiv preprint*.

[99] Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-218057.

[100] Mroczkowski, P. (2001). Implementation of the block cipher rijndael using altera fpga. *Journal of Telecommunications and Information Technology*, pages 80–86.

[101] Newport, E. L. (1990). Maturational constraints on language development. *Cognitive Science*, 14:11–28.

[102] NRC (2006). *Learning to Think Spatially*. Washington, D.C.: National Academies Press.

[103] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016). The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE.

[104] Patil, K. R., Zhu, X., Kopeć, $\mathscr{L}$., and Love, B. C. (2014). Optimal teaching for limited-capacity human learners. In *Advances in neural information processing systems*, pages 2465–2473.

[105] Peirce, C. S., Hartshorne, C., and Weiss, P. (1935). Collected papers of charles sanders peirce: (vol. i-vi).

[106] Plaut, D. C., McClelland, J. L., Seidenberg, M. S., and Patterson, K. (1996). Understanding Normal and Impaired Word Reading: Computational Principles in Quasi-Regular Domains. *Psychological Review*, 103(1):56–115.

[107] Qualtrics (2005). Qualtrics©2018. https://it.wisc.edu/services/surveys-qualtrics, last visited 01-18-2018.

[108] Queirolo, F. (2011). Steganography in images. *Final Communications Report.*, 3.

[109] Rau, M. A. (2017). Conditions for the effectiveness of multiple visual representations in enhancing stem learning. *Ed Psychology Review*, 29(4):717–761.

[110] Rau, M. A., Aleven, V., and Rummel, N. (2013). Interleaved practice in multi-dimensional learning tasks: Which dimension should we interleave? *Learning and Instruction*, 23(1):98–114.

[111] Rau, M. A., Aleven, V., and Rummel, N. (2015a). Successful learning with multiple graphical representations and self-explanation prompts. *Journal of Educational Psychology*, 107(1):30.

[112] Rau, M. A., Mason, B., and Nowak, R. D. (2016). How to model implicit knowledge? similarity learning methods to assess perceptions of visual representations. In *EDM*, pages 199–206.

[113] Rau, M. A., Michaelis, J. E., and Fay, N. (2015b). Connection making between multiple graphical representations: A multi-methods approach for domain-specific grounding of an intelligent tutoring system for chemistry. *Computers & Education*, 82:460–485.

[114] Rau, M. A., Zahn, M., Misback, E., and Burstyn, J. (2019). Adaptive support for representation skills in a chemistry its is more effective than static support. In *International Conference on Artificial Intelligence in Education*, pages 432–444. Springer.

[115] Reisberg, D. (2013). *The Oxford handbook of cognitive psychology*. Oxford University Press.

[116] Rensink, R. A. (2002). Change detection. *Annual Review of Psychology*, 53(1):245–277.

[117] Reyzin, L. and Russell, S. (2003). More efficient provably secure steganography. *Department of computer science Boston University*.

[118] Rich, E. and Knight, K. (1991). Artificial intelligence. *McGraw-Hill, New*.

[119] Richman, H. B., Gobet, F., Staszewski, J. J., and Simon, H. A. (1996). Perceptual and memory processes in the acquisition of expert performance: The epam model. *The road to excellence: The acquisition of expert performance in the arts and sciences, sports, and games*, pages 167–187.

[120] Rivest, R. L., Adleman, L., and Dertouzos, M. L. (1978a). On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180.

[121] Rivest, R. L., Shamir, A., and Adleman, L. (1978b). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.

[122] Rohde, D. L. and Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 72(1):67–109.

[123] Rohrer, D. and Taylor, K. (2007). The shuffling of mathematics problems improves learning. *Instructional Science*, 35(6):481–498.

[124] Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.

[125] Salamati, M., Soudjani, S., and Majumdar, R. (2019). Perception-in-the-loop adversarial examples. *arXiv preprint arXiv:1901.06834*.

[126] Sallee, P. (2003). Model-based steganography. In *International Workshop on Digital Watermarking*, pages 154–167. Springer.

[127] Sarkar, S., Bansal, A., Mahbub, U., and Chellappa, R. (2017). Upset and angri: Breaking high performance image classifiers. *arXiv preprint arXiv:1707.01159*.

[128] Schnotz, W. (2014). An integrated model of text and picture comprehension. *The Cambridge handbook of multimedia learning (2 ed., pp. 72-103)*.

[129] Seidenberg, M. S. (2017). *Language at the speed of sight: How we read, why so many can't, and what can be done about it*. Basic Books, New York.

[130] Seidenberg, M. S. and McClelland, J. L. (1989). A Distributed, Developmental Model of Word Recognition and Naming. *Psychological Review*, 96(4):523–568.

[131] Sen, A., Patel, P., Rau, M. A., Mason, B., Nowak, R., Rogers, T. T., and Zhu, X. (2018). Machine beats human at sequencing visuals for perceptual-fluency practice. *International Educational Data Mining Society*.

[132] Shanks, D. R. (2005). Implicit learning. *Handbook of cognition*, pages 202–220.

[133] Sharif, M., Bauer, L., and Reiter, M. K. (2018). On the suitability of lp-norms for creating and preventing adversarial examples. In *The Bright and Dark Sides of Computer Vision: Challenges and Opportunities for Privacy and Security (CVPR Workshop)*.

[134] Sheikh, H. R., Sabir, M. F., and Bovik, A. C. (2006). A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451.

[135] Siegelman, N., Kearns, D. M., and Rueckl, J. G. (2020). Using information-theoretic measures to characterize the structure of the writing system: the case of orthographic-phonological regularities in English. *Behavior Research Methods*.

[136] Sievert, S., Ross, D., Jain, L., Jamieson, K., Nowak, R., and Mankoff, R. (2017). Next: A system to easily connect crowdsourcing and adaptive data collection. In *Proceedings of the 16th Python in Science Conference*, pages 113–119.

[137] Simmons, G. J. (1984). The prisoners' problem and the subliminal channel. In *Advances in Cryptology*, pages 51–67. Springer.

[138] Simons, D. J. and Ambinder, M. S. (2005). Change blindness: Theory and consequences. *Current Directions in Psychological Science*, 14(1):44–48.

[139] Singla, A., Bogunovic, I., Bartók, G., Karbasi, A., and Krause, A. (2014). Near-optimally teaching the crowd to classify. In *ICML*, pages 154–162.

[140] Smart, N. P. and Vercauteren, F. (2010). Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography*, pages 420–443. Springer.

[141] Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *Journal of machine learning research*, 2(Nov):67–93.

[142] Su, J., Vargas, D. V., and Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*.

[143] Suh, J., Zhu, X., and Amershi, S. (2016). The label complexity of mixed-initiative classifier training. In *International Conference on Machine Learning*, pages 2800–2809.

[144] 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.

[145] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

[146] Tan, K. M., Killourhy, K. S., and Maxion, R. A. (2002). Undermining an Anomaly-Based Intrusion Detection System Using Common Exploits. In *Recent Advances in Intrusion Detection*.

[147] Thompson, A. (2017). All the news. *https://www.kaggle.com/snapcrack/all-the-news*.

[148] Uttal, D. H. and Doherty, K. O. (2008). Comprehending and learning from 'visualizations': A developmental perspective. *Visualization: Theory and practice in science education*, pages 53–72.

[149] Valizadeh, A. and Wang, Z. J. (2011). Correlation-and-bit-aware spread spectrum embedding for data hiding. *IEEE Transactions on Information Forensics and Security*, 6(2):267–282.

[150] Van Gog, T. and Paas, F. (2008). Instructional efficiency: Revisiting the original construct in educational research. *Educational Psychologist*, 43(1):16–26.

[151] Van Tilborg, H. C. and Jajodia, S. (2014). *Encyclopedia of cryptography and security*. Springer Science & Business Media.

[152] VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.

[153] Vernam, G. S. (1926). Cipher printing telegraph systems: For secret wire and radio telegraphic communications. *Journal of the AIEE*, 45(2):109–115.

[154] Vousden, J. I., Ellefson, M. R., Solity, J., and Chater, N. (2011). Simplifying reading: Applying the simplicity principle to reading. *Cognitive Science*, 35(1):34–78.

[155] Wang, Z. and Bovik, A. C. (2009). Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117.

[156] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on Image Processing*, 13(4):600–612.

[157] Wise, J. A., Kubose, T., Chang, N., Russell, A., and Kellman, P. J. (2003). Perceptual learning modules in mathematics and science instruction. *Teaching and learning in a network world'(IOS Press, 2000)*, pages 169–176.

[158] Wolfe, J. M. (2010). Visual search. *Current Biology*, 20(8):R346–R349.

[159] Wu, D.-C. and Tsai, W.-H. (2003). A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters*, 24(9-10):1613–1626.

[160] Wu, H.-C. (2007). The karush–kuhn–tucker optimality conditions in an optimization problem with interval-valued objective function. *European Journal of Operational Research*, 176(1):46–59.

[161] Wylie, R. and Chi, M. T. (2014). The self-explanation principle in multimedia learning. *R. E. Mayer (Ed.), The Cambridge handbook of multimedia learning*, pages 413–432.

[162] Xiao, C., Zhu, J.-Y., Li, B., He, W., Liu, M., and Song, D. (2018). Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*.

[163] Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

[164] Yuan, X., He, P., Zhu, Q., and Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*.

[165] Zhang, L., Wu, J., and Zhou, N. (2009). Image encryption with discrete fractional cosine transform and chaos. In *Information Assurance and Security, 2009. IAS'09. Fifth International Conference on*, volume 2, pages 61–64. IEEE.

[166] Zhang, X., Lin, W., and Xue, P. (2008). Just-noticeable difference estimation with pixels in images. *Journal of Visual Communication and Image Representation*, 19(1):30–41.

[167] Zhang, X., Zhu, X., and Lessard, L. (2019). Online data poisoning attack. *arXiv preprint arXiv:1903.01666*.

[168] Zhang, X., Zhu, X., and Wright, S. (2018). Training set debugging using trusted items. AAAI.

[169] Zhao, Z., Dua, D., and Singh, S. (2017). Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.

[170] Zhou, K., Doyle, J. C., Glover, K., et al. (1996). *Robust and optimal control*, volume 40. Prentice hall New Jersey.

[171] Zhu, X. (2013). Machine teaching for bayesian learners in the exponential family. In *Advances in Neural Information Processing Systems*, pages 1905–1913.

[172] Zhu, X. (2015). Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pages 4083–4087.

[173] Zhu, X., Liu, J., and Lopes, M. (2017). No learner left behind: On the complexity of teaching multiple learners simultaneously. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3588–3594. AAAI Press.

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