

# Learning with Weak Supervision from Physics and Data-Driven Constraints

Hongyu Ren, Russell Stewart, Jiaming Song, Volodymyr Kuleshov, Stefano Ermon

■ In many applications of machine learning, labeled data is scarce and obtaining additional labels is expensive. We introduce a new approach to supervising learning algorithms without labels by enforcing a small number of domain-specific constraints over the algorithms' outputs. The constraints can be provided explicitly based on prior knowledge — for example, we may require that objects detected in videos satisfy the laws of physics — or implicitly extracted from data using a novel framework inspired by adversarial training. We demonstrate the effectiveness of constraint-based learning on a variety of tasks — including tracking, object detection, and human pose estimation — and we find that algorithms supervised with constraints achieve high accuracies with only a small number of labels, or with no labels at all in some cases.

In many applications of machine learning — including image recognition (Krizhevsky, Sutskever, and Hinton 2012), machine translation (Sutskever, Vinyals, and Le 2014), and speech recognition (Graves, Mohamed, and Hinton 2013) — large labeled data sets are a key component for building state-of-the-art systems. Collecting such data sets can be expensive, representing a major bottleneck in deploying machine learning algorithms.

Humans, on the other hand, are able to learn most tasks without direct examples, opting instead for high-level instructions for how each task should be performed, or what it will look like when completed. In this work, we ask whether a similar principle can be applied to teaching machines: can we supervise algorithms with a few (or no) labeled examples by instead only describing how desired outputs should look or by giving a small set of examples of outputs?

Contemporary methods for learning with fewer labels are often based on semisupervised or unsupervised feature learning (Figueiredo and Jain 2002; Coates, Ng, and Lee 2011; Kingma et al. 2014). In this work, we instead propose to describe the desired behavior of a machine learning algorithm using constraints over its outputs (Shcherbatyi and Andres 2016). For example, we may require that the movements of an object tracked in a video satisfy the laws of physical mechanics. Unlike labels — which only apply to their corresponding inputs — constraints are specified once for the entire data set, providing an opportunity for more cost-effective supervision; moreover, a single set of constraints can be applied to multiple data sets without relabeling. This form of supervision has shown promising results in robotics (Mnih et al. 2015), preference learning (Choi, Van den Broeck, and Darwiche 2015), and semantic segmentation (Owens et al. 2011).

Supervising algorithms using constraints, which amounts to a very general form of weak supervision, represents a significant departure from standard supervised learning approaches. In this article, we formalize two concrete ways of defining constraints and using them to guide the behavior of machine learning systems.

### Learning with Explicit Constraints

In the explicit approach, constraints are handcrafted by humans based on prior domain knowledge. For example, humans can specify logical rules or physical laws governing the output of a system. As long as these invariants can be succinctly represented using mathematical formulas, the effort of specifying them does not increase with data set size. Furthermore, we may often reuse common invariants across related data sets.

As a concrete example, consider the problem of tracking the height of an object in free fall within multiple consecutive video frames. Clearly, the heights of the object in each frame are not independent, and their sequence demonstrates a well-defined structure. In fact, we know from elementary physics that any correct sequence of outputs forms a parabola. In our experiments, we find that requiring an object tracking system to output trajectories consistent with a parabola prescribed by physical mechanics can almost entirely replace the need to manually label each video frame and thus substantially reduce labeling efforts.

### Learning Constraints Implicitly from Data

However, specifying constraints by hand may require significant domain expertise and the constraints themselves may not always possess simple mathematical descriptions. In such cases, we propose to learn the invariants implicitly from a small set of representative output samples. These samples do not need to be tied to corresponding inputs (as in super-

vised learning) and may come from a black-box simulator that abstracts away physics-based formulas, examples of outputs collected by humans, and outputs extracted from standard data sets used in supervised learning.

Inspired by recent advances in generative modeling, we capture the distribution of outputs using an approach based on adversarial learning (Goodfellow et al. 2014). Specifically, we train two distinct learners: a primary model for the task at hand and an auxiliary classification algorithm called a discriminator. During training, we constrain the main model such that its outputs cannot be distinguished by the discriminator from true output samples, thus forcing the discriminator to capture the structure of the output space. This approach forms a novel adversarial framework for performing weak supervision with learned constraints.

In the aforementioned object tracking example, we may empirically measure trajectories of falling objects and fit the tracking system to produce outputs that cannot be distinguished from real trajectories by the auxiliary discriminator. In the process, the discriminator effectively learns the physical laws that govern falling objects and imposes them upon the tracking system. Although in this example we could have specified these laws by hand, the learning approach becomes essential when invariants cannot be specified succinctly, such as when training a joint detector that needs to satisfy the anatomical constraints imposed by the human body.

### Applications to Semisupervised Learning

Although constraint learning does not require a fully labeled data set containing input-label pairs, providing it with such data turns our problem into an instance of semisupervised learning. In semisupervised learning, we are given a small set of labeled examples and a large unlabeled data set. The goal is to use the unlabeled data to improve supervised accuracy.

In this setting, our constraint learning approach uses the small labeled data set to discover the high-level invariants governing the system's outputs and then uses these invariants to train the system on the large unlabeled set. This process can be interpreted as a new semisupervised algorithm for structured prediction problems. Experimental results demonstrate that this form of constraint learning performs well on a variety of semisupervised problems, such as tracking, object detection, and pose estimation, outperforming natural baselines with very few labeled inputs.

### Contributions

The primary contribution of this work is to introduce and formalize the notion of learning with constraints, and to demonstrate that this approach significantly reduces labeling effort across several structured pre-

diction tasks. We introduce two distinct paradigms for learning with constraints and show how they may be used to supervise learning algorithms, particularly modern methods based on deep neural networks. The frameworks of the two paradigms are shown in figures 1 and 2. A special case of our framework is a new approach to semisupervised learning. Our results are based on work by Stewart and Ermon (2017).

## Background

Our work focuses on structured prediction problems, and uses implicit generative models to learn constraints. In this section, we introduce structured prediction and implicit models. The following sections explore various forms of constraint learning.

### Supervised Learning and Structured Prediction

In supervised learning, we are given a training set of  $n$  examples, where each example includes an input  $x_i$  (that belongs to  $X$ ) and the corresponding label  $y_i$  (that belongs to  $Y$ ). We learn a function  $f$ , mapping inputs to labels by minimizing a loss  $l$  within a hypothesis class  $H$ . By restricting the space of possible functions to  $H$ , we are leveraging prior knowledge about the specific problem we are trying to solve. Alternatively, we may incorporate prior knowledge by specifying an apriori preference for certain functions in the hypothesis class via a regularization term  $R(f)$ . Together, these concepts give rise to the supervised learning objective

$$f^* = \arg \min_{f \in H} \sum_{i=1}^n l(f(x_i), y_i) + R(f). \quad (1)$$

In this work, we consider the class of functions  $H$  parameterized by convolutional neural networks. We are also interested in structured prediction problems (Koller and Friedman 2009), where the outputs  $y_i$  are complex vector-valued objects with strong correlation among their components. Examples of structured outputs include vectors, trees, sequences, and graphs (Taskar et al. 2005; Daumé, Langford, and Marcu 2009).

### Adversarial Training and Implicit Probabilistic Models

Adversarial training is a particular technique for fitting structured prediction models. It is most widely used to train implicit probabilistic models (Mohamed and Lakshminarayanan 2016). Implicit models are defined as the result of a stochastic sampling procedure, rather than through an explicitly defined likelihood function. A prominent example is generative adversarial networks (GAN), in which samples are obtained by transforming Gaussian noise via a neural network  $G$ , called the generator.

In this work, we will be interested in placing constraints on a probability distribution over the output

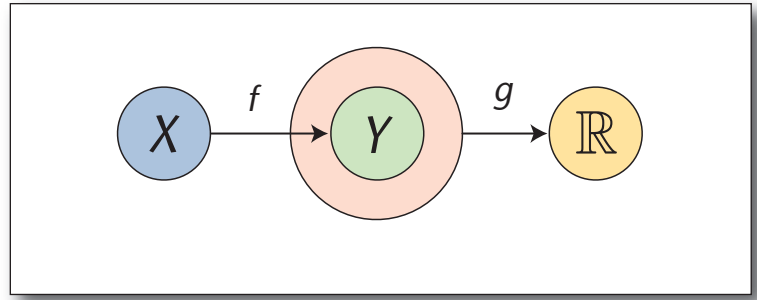


Figure 1. Framework of the Paradigm for Learning with Explicit Constraints.

Explicit constraints are algebraic or logical formulas that hold over the output space  $Y$  and are specified based on prior domain knowledge.

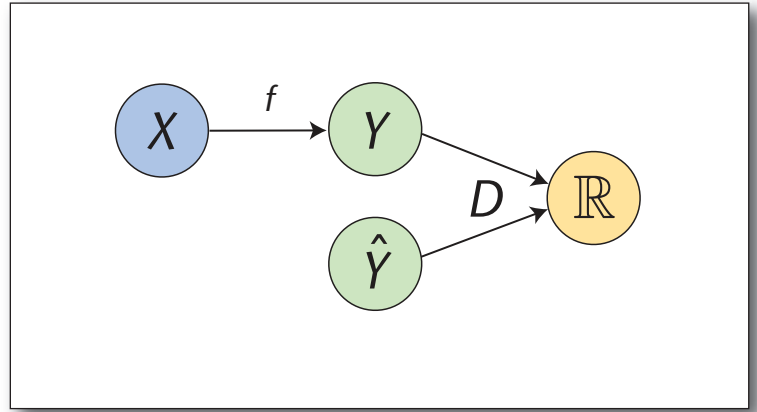


Figure 2. Framework of the Paradigm for Learning Implicitly from the Data.

Constraints are learned implicitly from data by forcing  $f$  to produce outputs that are indistinguishable from representative outputs  $Y$  by an auxiliary discriminator  $D$ .

space  $Y$ . We are going to define this distribution implicitly by a sampling procedure that samples an input and generates a label using a neural network. Note that evaluating the likelihood of the model defined by this sampling procedure is typically intractable.

Adversarial training of implicit models possesses two interpretations. The first is that of a mathematical game, in which the generator  $G$  tries to fool a discriminator  $D$  from distinguishing generated samples from real samples. This process results in a minimax objective, which can be optimized through stochastic gradient descent. Alternatively, this process can be interpreted as minimizing distances or divergences between distributions, which include approximations of the Jensen-Shannon divergence (Goodfellow et al. 2014), the Earth Mover's distance (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017), or differences in statistics between samples from two distributions (Li, Swersky, and Zemel 2015).

## Learning with Explicit Constraints

The goal of constraint-based learning is to train a model  $f$ , mapping from inputs to outputs that we care about, using only high-level rules rather than labeled examples.

We focus on structured prediction problems, in which the output is a vector  $y$  with correlated components. For example,  $y$  may correspond to the trajectory of a falling object in a sequence of video frames  $x$ . Clearly, the heights in each frame are not independent, and the sequence demonstrates a well-defined structure defined by physics. We can utilize this physical law as a constraint and learn to detect the object without resorting to exhaustive labeling.

### Replacing Supervised Losses with Constraints

To enforce our prior knowledge of the structure of  $y$ , we specify a weighted constraint function  $g$ , which penalizes output structures that are not consistent with our understanding of the task. The key question we explore in this work is whether this weak form of supervision is sufficient to achieve high labeled accuracy on a test set.

While one clearly needs labels to evaluate the optimal function, labels may not be necessary to discover that optimal function. If prior knowledge informs us that outputs of the optimal function have other unique properties among functions in the hypothesis class, we may use these properties as constraints to train the system without the need for explicit labeled examples.

Specifically, we first consider an approach in which no labels are provided to us, and optimize for a necessary property of the output (the constraint) instead. That is, we search for the function that optimally satisfies the constraint requirements

$$f^* = \arg \min_{f \in H} \sum_{i=1}^n g(x_i, f(x_i)) + R(f) \quad (2)$$

In our experiments, we find that commonly used hypothesis classes (convolution layers encoding translation invariance) and simple regularization terms may be sufficient to avoid functions that optimize only for the constraint but not the original loss function. In these settings, we can optimize the constraint in place of the loss function with stochastic gradient descent (SGD), freeing us from the need for labels.

### Regularization for Constraint Learning

When optimizing for the constraint alone is not sufficient to find the desired solution, we may add additional regularization terms  $R(f)$  to supervise the machine towards correct convergence. For example, if the constraint is undesirably satisfied by a function that produces constant output at every frame, we add a term to favor outputs with higher entropy, leading

to the correct function. The process of designing the precise constraint and the regularization term is a form of supervision, and can require a significant time investment. But unlike hand labeling, it does not increase proportionally to the size of the training data set, and can be applied to new data sets often without modification.

## Adversarial Constraint Learning

In the sciences, discovering constraints is often a data-driven process — for example, the laws of physics are often discovered by validating hypotheses with experimental results before formulas are summarized.

Motivated by this idea, we ask ourselves whether we can learn constraints (such as physical laws) from data, rather than requiring that they be specified by humans. This approach enables us to apply constraint learning in settings in which the invariants governing a system's output are too complex to be specified manually.

### Learning Constraints from Data

Suppose that we are given a small number of outputs (that is, labels that are not necessarily associated with inputs) or a black-box mechanism/simulator for generating such outputs. We formulate the task of learning a constraint loss from these labeled samples using the framework of generative adversarial learning (Goodfellow et al. 2014).

Our ultimate goal is to learn a function  $f(x)$  that produces samples that lie close to the manifold of true output samples in  $Y$ . To enforce this goal, we follow the approach of Goodfellow et al. (2014) and define an auxiliary classifier  $D$  called a discriminator, which tries to assign higher scores to the real set of labels (since they follow the constraints by assumption) and lower scores to outputs from  $f(x)$ . At the same time, we train  $f(x)$  to produce outputs that score higher under the discriminator. Thus, the discriminator learns to effectively extract the constraints in the samples and impose them upon  $f(x)$ ; and since the goal of  $f(x)$  is to produce outputs that score high under the discriminator, this function learns to meet the desired constraints.

Figure 3 shows an overview of the adversarial constraint learning framework when the outputs form a trajectory from an object tracking system. The discriminator tries to distinguish generated trajectories (outputs from the function  $f(x)$ ) from real sample trajectories, while the regressor tries to output trajectories that match the distribution provided by a black-box simulator. When trained to optimality (and assuming both models have enough capacity), the discriminator represents the implicit constraint while the regressor learns to perform structured prediction that satisfies this constraint.

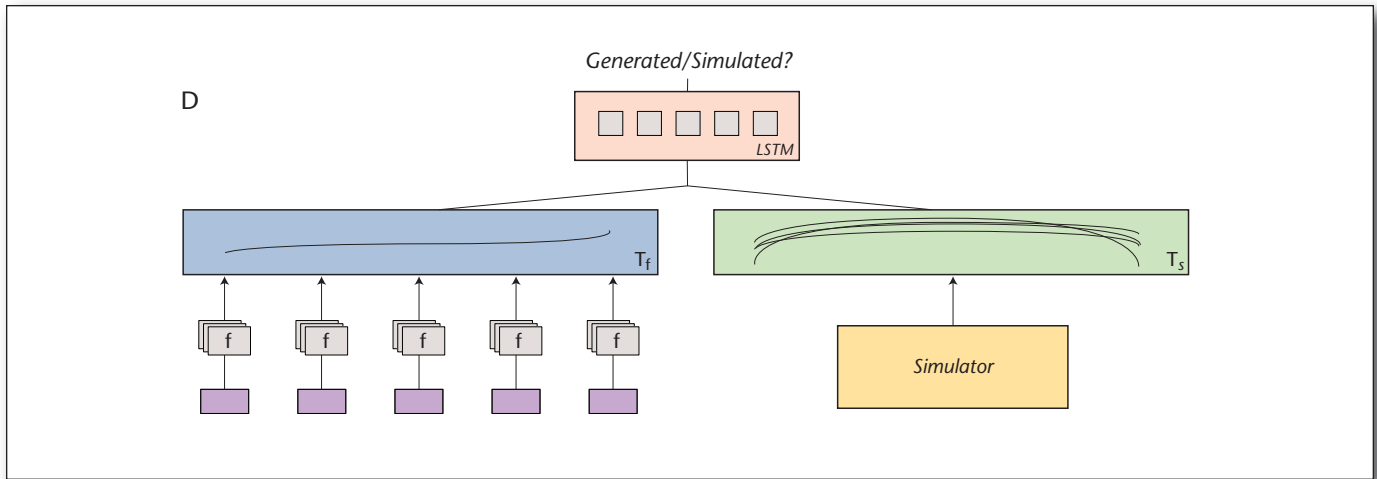


Figure 3. Adversarial Constraint Learning for an Object Tracking System.

We train the function  $f$  by asking it to generate trajectories  $T_f$  of a moving object that cannot be discriminated from sample trajectories  $T_s$ .

### Constraint Learning by Matching Distributions

Our approach to learning constraints in a generative adversarial framework can also be interpreted as matching the distribution over labels defined by the model to the true marginal data distribution over the labels. The former is specified implicitly by a sampling mechanism in which we first take a random input  $x$  and obtain an output  $y$  by passing it through a deterministic function  $f(x)$  (for example, a neural network that outputs a sequence of object positions given a set of input frames). Matching marginal distributions over the labels is a necessary condition for the correct model, and it can be interpreted as a form of regularization over the output space  $Y$ .

In slightly more formal terms, our constraint learning framework can be seen as optimizing a measure of similarity between distributions that can be approximately computed through samples. Examples of such similarity measures include the Jensen-Shannon divergence or the Earth Mover's distance. Minimizing these divergences or distances is equivalent to training the model to satisfy the constraints implicitly encoded in a set of representative output samples.

### Semisupervised Structured Prediction

Although constraint learning does not require a fully labeled data set containing input-label pairs, providing it with such data turns our problem into an instance of semisupervised learning. In semisupervised learning, we are given a small set of labeled examples and a large unlabeled data set. The goal is to use the unlabeled data to improve supervised accuracy.

Our constraint learning approach uses the small labeled data set to discover the high-level invariants governing the system's outputs and then uses these

invariants to train the system on the large unlabeled set. In addition, we may combine our constraint learning objective with a standard classification loss term (over the labeled data), which acts as an additional regularizer. This process can be interpreted as a new semisupervised algorithm for structured prediction problems, such as tracking, object detection, and pose estimation.

Traditional semisupervised learning methods assume there is a large source of inputs  $x$  and tend to impose regularization over the input space. Our method, on the other hand, can exploit abundant samples from the output space that are not matched to particular inputs. Moreover, our method can be easily combined with other approaches (Kingma et al. 2014; Li, Zhu, and Zhang 2016; Salimans et al. 2016; Miyato et al. 2017) to further boost performance.

## Experiments

We perform four experiments that demonstrate the effectiveness of constraint learning in various real-world settings. We refer to the trained model as a regression network (or simply as a regressor)  $f$ , mapping from inputs to outputs that we care about.

Our first two experiments use explicit constraints in the form of formulas; the latter two rely on adversarial constraint learning, where we train an auxiliary discriminator using output samples from a black-box simulator. We refer the readers to our papers for network layout and training details (Stewart and Ermon 2017).

### Tracking an Object in Free Fall

In our first experiment, we record videos of an object being thrown across the field of view and aim to learn the object's height in each frame. Example



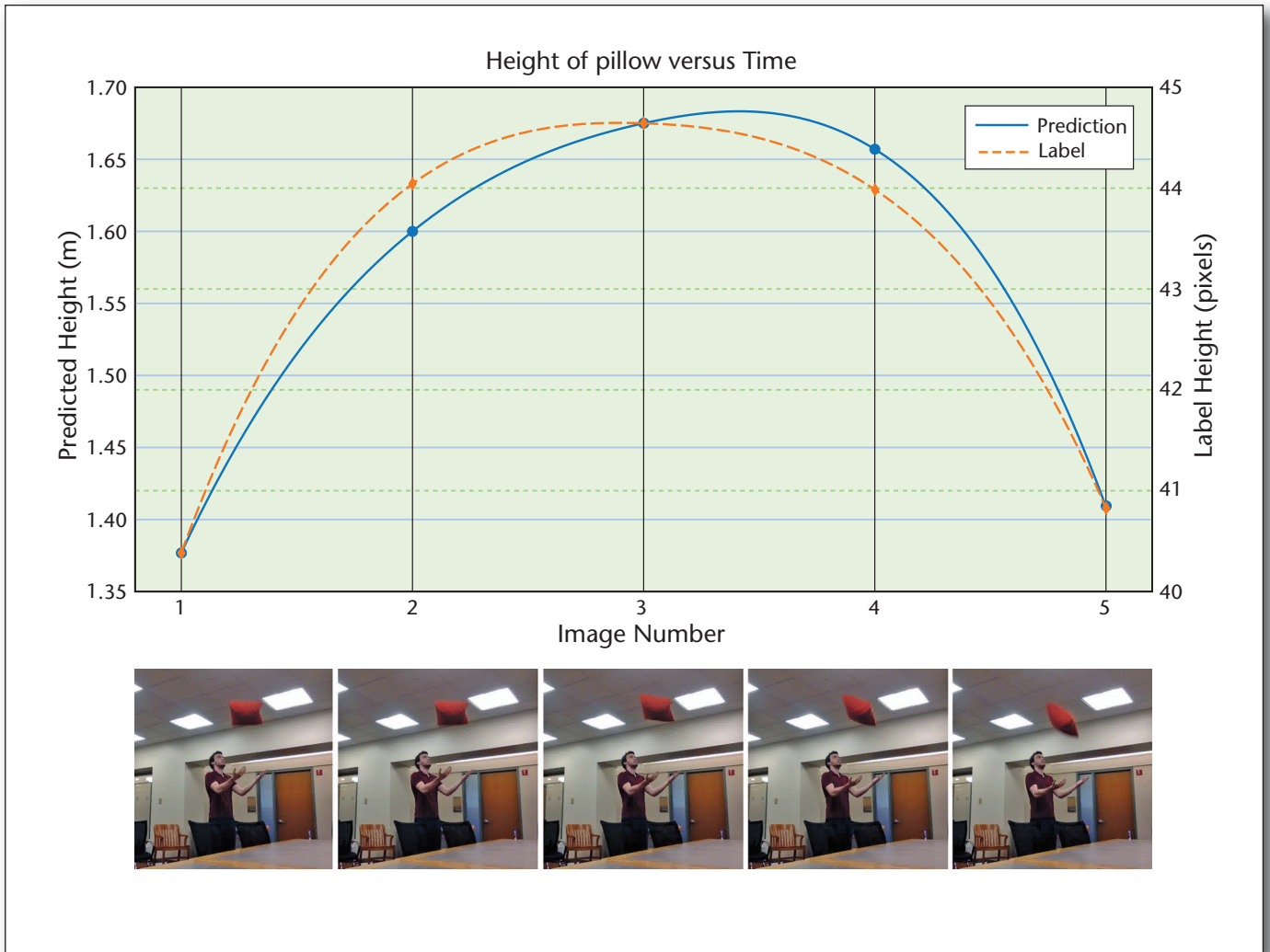


Figure 4. Qualitative Results from Our Network Applied to Fresh Images.

As the pillow is tossed, the height forms a parabola over time. We exploit this structure to independently predict the pillow's height in each frame without providing labels.

frames are shown in figure 4. Our goal is to obtain a regression network on color images, that is, a mapping from images to a real number. We will train this network as a structured prediction problem operating on a sequence of  $N$  images to produce a sequence of  $N$  heights, and each piece of data  $x_i$  will be a vector of images,  $x$ . Rather than supervising our network with direct labels,  $y$ , we instead supervise the network to find an object obeying the elementary physics of free-falling objects. Because gravity acts equally on all objects, we need not encode the object's mass or volume.

#### Constraints

An object acting under gravity will have a fixed acceleration of  $a = -9.8 \text{ m/s}^2$ , and the plot of the object's height over time will form a parabola:

$$y_i = y_0 + v_0(i\Delta t) + a(i\Delta t)^2 \quad (3)$$

where  $\Delta t = 0.1\text{s}$  is the duration between frames and  $y_0$  and  $v_0$  denote the initial location and velocity respectively. This equation provides a necessary constraint, which the correct mapping  $f^*$  must satisfy. We thus train  $f$  by making incremental improvements in the direction of better satisfying this equation.

Given any trajectory of  $N$  height predictions,  $f(x)$ , we fit a parabola with fixed curvature to those predictions, and minimize the constraint loss, which is the residual between the predictions and the parabola. Because the constraint loss is differentiable almost everywhere, we can optimize it with SGD. Surprisingly, we find that when combined with existing reg-

ularization methods for neural networks, this optimization is sufficient to recover  $f^*$  up to an additive constant  $C$  (specifying what object height corresponds to 0). Qualitative results from our network applied to fresh images after training are shown in figure 4.

#### Evaluation

We manually label the height of our falling objects in pixel space. Note that labeling the true height in meters requires knowing the object's distance from the camera, so we instead evaluate by measuring the correlation of predicted heights with ground truth pixel measurements. All results are evaluated on test images not seen during training. Note that a uniform random output would have an expected correlation of 12.1 percent. Our network results in a correlation of 90.1 percent. For comparison, we also train a supervised network on the labels to directly predict the height of the object in pixels. This network achieves a correlation of 94.5 percent, although this task is somewhat easier as it does not require the network to compensate for the object's distance from the camera.

This experiment demonstrates that one can teach a neural network to extract object information from real images by writing down only the equations of physics that the object obeys.

### Detecting Objects with Causal Relationships

In addition to physics, other sources of domain knowledge can in principle be used to provide supervision in the learning process. For example, significant efforts have been devoted in the past few decades to construct large knowledge bases (Lenat 1995; Bolacker et al. 2008). This knowledge is typically encoded using logical- and constraint-based formalisms. Thus, in our next experiment, we explore the possibilities of learning from logical constraints imposed on single images. More specifically, we ask whether it is possible to learn from causal phenomena.

We provide our model images containing a stochastic collection of up to four characters: Peach, Mario, Yoshi, and Bowser, with each character having small appearance changes across frames due to rotation and reflection. Example images can be seen in figure 5. While the existence of objects in each frame is nondeterministic, the generating distribution encodes the underlying phenomenon that Mario will always appear whenever Peach appears. Our aim is to create a pair of neural networks  $f = (f_1, f_2)$  for identifying Peach and Mario, respectively. The networks,  $f_1$  and  $f_2$ , map the image to the discrete Boolean variables,  $y_1$  and  $y_2$ . This problem is challenging because the networks must simultaneously learn to recognize the characters and select them according to logical relationships.

#### Constraints

Rather than supervising with direct labels, we train

the networks by constraining their outputs to have the logical relationship  $y_1 \Rightarrow y_2$ , which means if  $y_1$  is true (1),  $y_2$  can never be false (0). However, merely satisfying the constraint  $y_1 \Rightarrow y_2$  is not sufficient to certify learning. For example, the system might falsely report the constant output,  $y_1 \equiv 1, y_2 \equiv 1$  on every image. Such a solution would satisfy the constraint, but say nothing about the presence of characters in the image.

To avoid such trivial solutions, we add three loss terms. The first loss forces rotational independence of the output by applying a random horizontal and vertical reflection,  $\rho$ , to images and requiring the network output to be the same. This encourages the network to focus on the existence of objects, rather than location. The second and third loss allow us to avoid trivial solutions by encouraging high standard deviation and high entropy outputs across a input batch of 16 images, respectively.

Even with these constraints, the loss remains invariant to logical permutations (for example, given a correct solution  $(y_1^*, y_2^*)$ , the incorrect solution  $(y_1', y_2')$  would satisfy  $y_1' \Rightarrow y_2'$ , and have the same entropy).

$$\begin{aligned} y_1' &= y_1^* \\ y_2' &= (y_1^* \wedge y_2^*) \vee (\neg y_1^* \wedge \neg y_2^*) \end{aligned} \quad (4)$$

We address this issue by forcing each Boolean output to derive its value from a single region of the image (each character can be identified from a small region in the image.) The Peach network,  $f_1$ , runs a series of convolution and pooling layers to reduce the original input image to a  $7 \times 7 \times 64$  grid. We find the 64-dimensional spatial vector with the greatest mean and use the information contained in it to predict the first binary variable. Examples of channel means for the Mario and Peach networks can be seen in figure 5. The Mario network,  $f_2$ , performs the same process. But if the Peach network claims to have found an object,  $f_2$  is prevented from picking any vector within two spaces of the location used by the first vector.

#### Evaluation

On a test set of 128 images, the network learns to map each image to a correct description of whether the image contains Peach and Mario. This experiment demonstrates that networks can learn from constraints that operate over discrete sets with potentially complex logical rules. Removing additional constraints will cause learning to fail. Thus, the experiment also shows that sophisticated sufficiency conditions can be key to success when learning from constraints.

### Pendulum Tracking

For this task, we downloaded a video of a pendulum from YouTube,<sup>1</sup> and we ask whether it is possible to extract the angle of the pendulum over time. Given

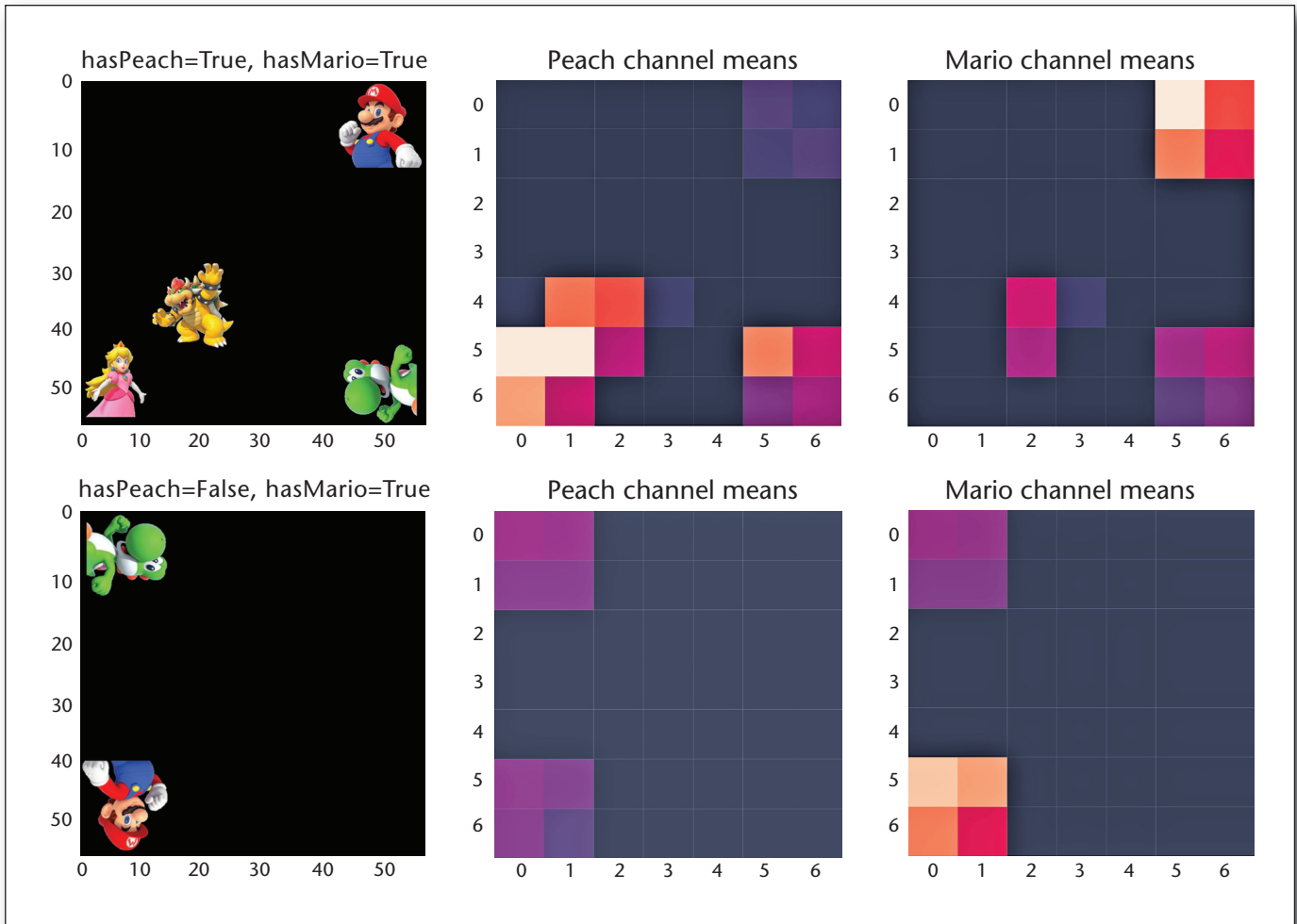


Figure 5. Example Images.

Whenever Peach (blond) shows up, Mario (red) comes around, but not vice versa. Yoshi (green) and Bowser (orange) appear randomly. The system trains with this high-level knowledge and learns to answer whether each image contains Peach or Mario. The first column contains example images. The second and third columns show the attended locations for the Peach and Mario networks, respectively.

an input image, we try to train a regression neural network to output the angle of the pendulum in the input image. This is also the first setting where we learn constraints implicitly.

#### Constraints

Since the outputs of the regressor over continuous frames should form a sine wave, we can provide a simulator that generates reasonable samples of trajectories. We define the structured prediction problem by concatenating the network outputs of contiguous images and form a high-dimensional trajectory. Unlike previous experiments, no explicit formulas are given throughout the experiment, and the (implicit) constraints are learned by the discriminator, using samples provided by the simulator.

#### Evaluation

After 5000 updates, the regressor converges to rela-

tively stable predictions for each frame. We then manually label the angle of the ball of the pendulum in each frame in the test set, and measure the correlation of the predicted position with the ground truth label in pixels. We achieve a correlation of 96.3 percent. Example predictions on the test data are shown in figure 6. In this experiment, while the formula-driven approach is arguably more appropriate for this problem, we demonstrate that our model is nonetheless capable of solving the task by learning the constraints implicitly through experience with samples from a black-box simulator.

#### Tracking Two Pendulums Simultaneously

To test the capability of our model to deal with more complex dynamics, we present synthetic images that contain two pendulums, and aim to track both of them. The two pendulums are independent, as



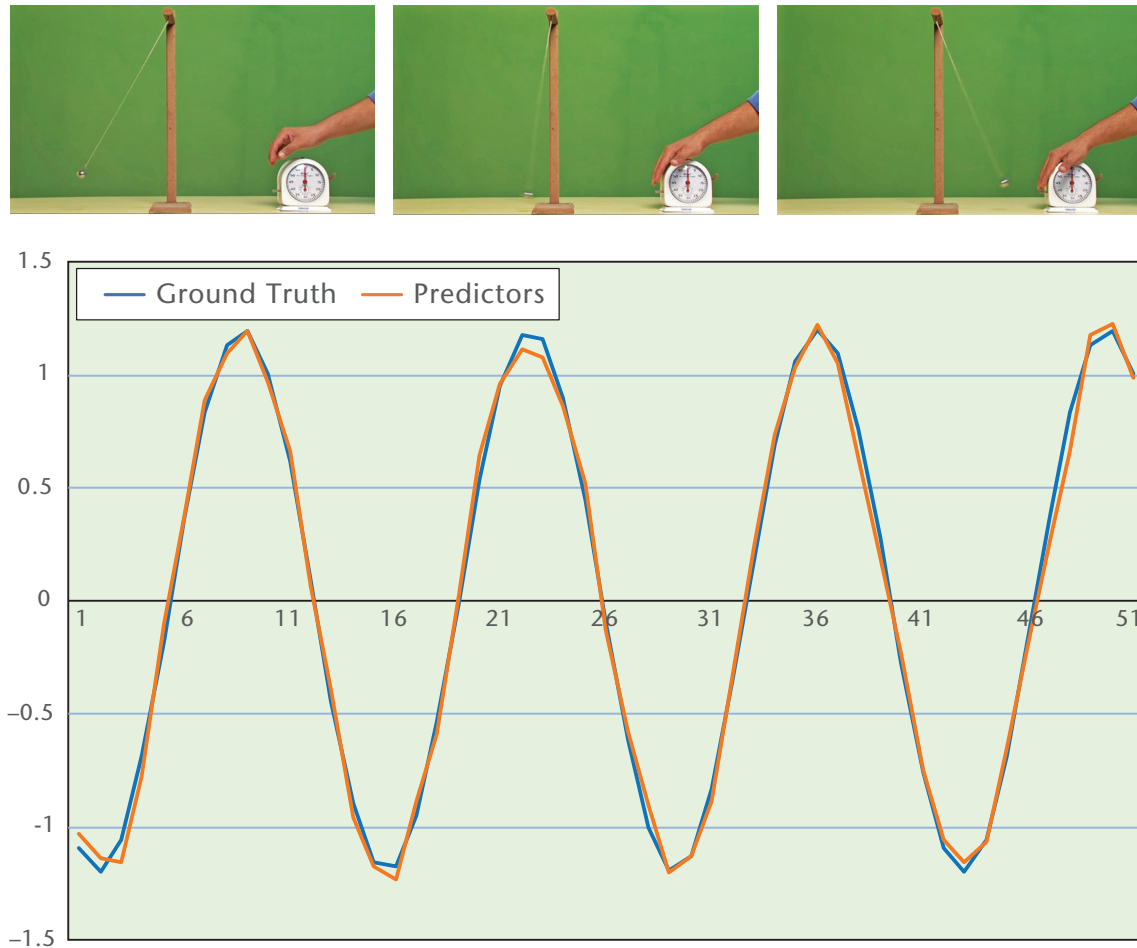


Figure 6. Example Predictions on the Test Data.

*Top:* frames from video used in the pendulum experiment. *Bottom:* the network is trained to predict angles that cannot be distinguished from the simulated dynamics, encouraging it to track the metal ball over time.

shown in figure 7. The regressor takes in an image and outputs two numbers, representing the angle of both pendulums. Similar to the pendulum-tracking experiments, the trajectories are constructed by the outputs of the regressor across 10 continuous images. We also provide a simulator of the joint dynamics of both pendulums. The model is thus trained adversarially when the discriminator tries to distinguish the outputs of the regressor and the simulated trajectories.

Our trained model achieves an average correlation of 99.2 percent between the predicted angles and the ground truth angles for detecting both pendulums. Note that the regressor will not converge to tracking only one pendulum with both outputs. Although such a situation may occur early in training, the dis-

criminator quickly learns to distinguish the correlated joint trajectories (if the regressor outputs two same numbers) from the independent joint trajectories (where the two numbers are independent), and the adversarial loss forces the regressor to track both pendulums.

Overall, the real-world pendulum experiment shows that using adversarial constraint learning it is possible to train a neural network to extract object information from real images using only a simulator of physics that the object obeys.

### Pose Estimation

In this experiment, we benchmark the proposed model on pose estimation, which has a larger output space. We aim to learn a regression network, map-

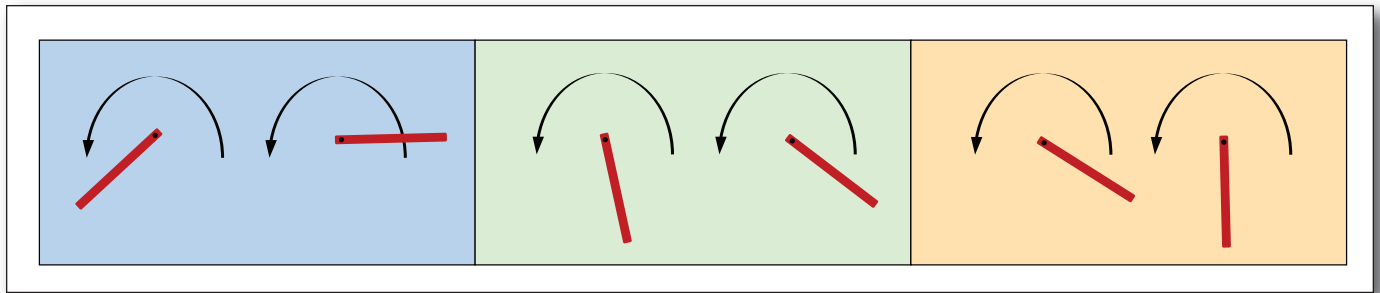


Figure 7. Two Pendulums.

We simulated two independent pendulums and aim to train a regression network to track both at the same time.

ping images to  $k \times 2$  real numbers, where  $k$  denotes the number of joints we detect, each having two coordinates. As before, we train the network based on a sequence of images, output a sequence of joint locations, and form a trajectory, which should be indistinguishable from the sample data.

The experiment is performed on a CMU multi-modal action database (MAD) (Huang et al. 2014). MAD contains 40 videos of 20 subjects (2 for each subject) performing a sequence of 35 actions in each video. We edit the 40 videos, extract the frames when the subjects perform the Jump and Side-Kick action, and train a network to detect the location of left / right hip / knee / foot based on the edited frames. The processed data set contains 620 valid frames (40 groups).

#### Constraints

In this scenario, consecutive outputs of the regressor should form a series of point sets; each point set should make up a skeleton that looks like a human; and the skeleton series should also perform the Jump and Side Kick action. Such constraints are difficult to express using mathematical formulas, and therefore we attempt to discover them through adversarial constraint learning. In this experiment, the simulator's outputs (real inputs for the discriminator) are actual labels  $\{\gamma_1, \dots, \gamma_n\}$ , but we assume we have no knowledge of the corresponding input vectors.

#### Evaluation

We use PCK@0.1 (Yang and Ramanan 2013) for evaluation. The prediction is considered correct if and only if it lies within  $\alpha \max(h, w)$  pixels from the correct location, where  $h$  and  $w$  denote the height and width of the tightest bounding box that covers the whole body, and we use  $\alpha = 0.1$ , which is a fairly strict criterion.

As in the pendulum experiment, the regressor is applied to each frame independently, and no knowledge of the neighboring frames is used in this process. We concatenate the regressor's outputs for each group and pass them to the discriminator. The discriminator is LSTM based and tries to tell the predicted locations and sample joint locations apart.

The results are shown in table 1. When only trained adversarially (0%+adv), the network is able to find the correct shape of the joints for each frame, but the predictions are biased. Since the subjects are not strictly acting in the center of the image, a minor shift  $(\Delta x, \Delta y)$  for all predicted joint locations still meets the requirements imposed by the discriminator, which encodes the structure of the output space. Mere adversarial training (without any labeled examples) is not sufficient for this task. To mitigate this problem, we provide a small amount of labeled training data and consider the semisupervised objective. The label loss helps adjust the regressor to output the precise locations for all joints. Given just 25 percent of the available labeled data, the regressor converges to detecting the joints with high accuracy, as shown in table 1. 50%+adv achieves same performance as 100 percent (fully supervised on all the available data) on the detection of feet, which have large movements.

#### Semisupervised Learning with Constraints

We further prove the value of adversarial training by evaluating the following three baselines. First, we test the result of a random simulator sample, using a randomly picked label from the simulator for each test data as its prediction. We also run ablation experiments  $t\%$ , where only  $t\%$  of the labeled data is used for supervised learning. In this case, some data points are neither trained nor tested.  $t\%$ +adv generally shows much better results compared to  $t\%$ . Lastly, we test  $t\%$ +rand, where we randomly assign labels from the simulator to unlabeled data points, and then use supervised learning. Although this random label assignment is very likely to be incorrect, it could still provide some signal given that the output space is structured. However, the results demonstrate that if the remaining data is used in this random manner, the detection accuracy hardly increases, and the accuracy of detecting feet decreases sharply. This emphasizes the importance of our adversarial training loss.

Our pose estimation experiment demonstrates that our model can handle large output spaces.

PCK@0.1(%)	L Hip	L Knee	L Foot	R Hip	R Knee	R Foot
RSS	0.517	0.414	0.300	0.520	0.412	0.299
0%+rand	0.743	0.620	0.493	0.750	0.604	0.442
0%+adv	0.846	0.578	0.414	0.824	0.636	0.514
12.5%	0.820	0.794	0.717	0.813	0.729	0.625
12.5%+rand	0.789	0.623	0.498	0.819	0.598	0.464
12.5%+adv	0.857	0.831	0.783	0.939	0.823	0.668
25%	0.766	0.869	0.768	0.769	0.885	0.737
25%+rand	0.852	0.804	0.560	0.864	0.763	0.510
25%+adv	0.923	0.842	0.829	0.914	0.850	0.802
37.5%	0.912	0.884	0.813	0.913	0.897	0.796
37.5%+rand	0.896	0.714	0.591	0.899	0.743	0.579
37.5%+adv	0.944	0.916	0.858	0.951	0.898	0.867
50%	0.943	0.903	0.809	0.958	0.904	0.773
50%+rand	0.841	0.725	0.606	0.847	0.815	0.733
50%+adv	0.965	0.895	0.861	0.968	0.922	0.872
100%	0.994	0.950	0.876	0.994	0.977	0.858

Table 1. PCK@0.1 Results on MAD.

Random simulator sample (RSS) makes a prediction using a random label from the simulator (baseline).  $t\%$  means that we train on  $t\%$  of the labeled data (standard supervised learning).  $t\%$ +rand means that we additionally randomly assign labels from the simulator to the remaining  $(1 - t\%)$  of the training data (baseline).  $t\%$ +adv means that we use  $t\%$  of the labeled training data (supervised loss), with the additional adversarial loss (our approach). Our approach consistently outperforms the baselines.

Handcrafting formula-based constraints in such high-dimensional spaces is tedious and error prone. Our model instead extracts constraints implicitly from the output samples.

## Conclusion

In this article, we introduced a new approach to supervising machine learning algorithms using constraints. These constraints can be explicitly specified as formulas, or they can be encoded implicitly in a set of representative output samples and extracted via the intermediary of a discriminator neural network. These constraints can be used in conjunction with labels, which gives rise to a new technique for semisupervised learning. Experimental results in several tasks reflecting real-world applications demonstrate the effectiveness of learning with constraints. Our approach, therefore, has the potential to significantly reduce the burden of labeling large data sets with hundreds of millions of examples.

## Note

1. [www.youtube.com/watch?v=02w9ISii\\_Hs](http://www.youtube.com/watch?v=02w9ISii_Hs)

## References

- Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. Unpublished arXiv preprint. arXiv:1701.07875 [stat.ML]. Ithaca, NY: Cornell University Library.
- Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1247–1250. New York: Association for Computing Machinery. doi.org/10.1145/1376616.1376746
- Choi, A.; Van den Broeck, G.; and Darwiche, A. 2015. Tractable Learning for Structured Probability Spaces: A Case Study in Learning Preference Distributions. In *Proceedings of 24th International Joint Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press.
- Coates, A.; Ng, A.; and Lee, H. 2011. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 215–223. Edinburgh, UK: PMLR.
- Daumé H.; Langford, J.; and Marcu, D. 2009. Search-Based Structured Prediction. *Machine Learning* 75(3): 297–325. doi.org/10.1007/s10994-009-5106-x
- Figueiredo, M. A. T., and Jain, A. K. 2002. Unsupervised Learning of Finite Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(3): 381–396. doi.org/10.1109/34.990138
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, 2672–2680. December 8–13, Montreal, Quebec, Canada.

- Graves, A.; Mohamed, A.-R.; and Hinton, G. 2013. Speech Recognition with Deep Recurrent Neural Networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649. Piscataway, NJ: Institute for Electrical and Electronics Engineers. doi.org/10.1109/ICASSP.2013.6638947
- Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved Training of Wasserstein Gans. Unpublished arXiv preprint. arXiv:1704.00028 [cs.LG]. Ithaca, NY: Cornell University Library.
- Huang, D.; Yao, S.; Wang, Y.; and De La Torre, F. 2014. Sequential Max-Margin Event Detectors. In *13th European Conference on Computer Vision*, Lecture Notes on Computer Science 8689, 410–424. Berlin: Springer. doi.org/10.1007/978-3-319-10578-9\_27
- Kingma, D. P.; Mohamed, S.; Rezende, D. J.; and Welling, M. 2014. Semi-Supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems 27*, 3581–3589. December 8–13, Montreal, Quebec, Canada.
- Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: The MIT Press.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 26*, 1097–1105. December 5–8, 2013, Lake Tahoe, Nevada.
- Lenat, D. B. 1995. Cyc: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11): 33–38. doi.org/10.1145/219717.219745
- Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative Moment Matching Networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML'15)*, 1718–1727. Edinburgh, UK: PMLR.
- Li, C.; Zhu, J.; and Zhang, B. 2016. Max-Margin Deep Generative Models for (Semi) Supervised Learning. Unpublished arXiv preprint. arXiv:1611.07119 [cs.CV]. Ithaca, NY: Cornell University Library.
- Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2017. Virtual Adversarial Training: A Regularization Method for Supervised and Semisupervised Learning. Unpublished arXiv preprint. arXiv:1704.03976 [stat.ML]. Ithaca, NY: Cornell University Library.
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Demis, H.; et al. 2015. Human-Level Control Through Deep Reinforcement Learning. *Nature* 518(7540): 529–533. doi.org/10.1038/nature14236
- Mohamed, S., and Lakshminarayanan, B. 2016. Learning in Implicit Generative Models. Unpublished arXiv preprint. arXiv:1610.03483 [stat.ML]. Ithaca, NY: Cornell University Library.
- Owens, T.; Saenko, K.; Chakrabarti, A.; Xiong, Y.; Zickler, T.; and Darrell, T. 2011. Learning Object Color Models from Multi-View Constraints. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 169–176. Piscataway, NJ: Institute for Electrical and Electronics Engineers. doi.org/10.1109/CVPR.2011.5995705
- Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved Techniques for Training Gans. In *Advances in Neural Information Processing Systems 29*, 2234–2242. December 5–10, 2016, Barcelona, Spain.
- Shcherbatyi, I., and Andres, B. 2016. Convexification of Learning from Constraints. Unpublished arXiv preprint. arXiv:1602.06746 [cs.LG]. Ithaca, NY: Cornell University Library.
- Stewart, R., and Ermon, S. 2017. Label-Free Supervision of Neural Networks with Physics and Domain Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2576–2582. Palo Alto, CA: AAAI Press.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, 3104–3112. December 8–13, Montreal, Quebec, Canada.
- Taskar, B.; Chatalbashev, V.; Koller, D.; and Guestrin, C. 2005. Learning Structured Prediction Models: A Large Margin Approach. In *Proceedings of the 22nd International Conference on Machine Learning*, 896–903. New York: Association for Computing Machinery. doi.org/10.1145/1102351.1102464
- Yang, Y., and Ramanan, D. 2013. Articulated Human Detection with Flexible Mixtures of Parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(12): 2878–2890. doi.org/10.1109/TPAMI.2012.261
- Hongyu Ren** is a senior undergraduate student from the School of Electronics Engineering and Computer Science, Peking University, China. He is interested in deep generative models and their applications in semisupervised/unsupervised learning settings. Currently, he is visiting Stanford University as a research scholar.
- Russell Stewart** is a PhD student in the Department of Computer Science at Stanford University. He is advised by Stefano Ermon. His research focuses on supervising neural networks without labels. Previously, he worked as a software engineer at Microsoft on the Kinect and as a research advisor for MetaMind and Mathpix.
- Jiaming Song** is a second-year PhD student at Stanford University. He is advised by Stefano Ermon. His current research interests are deep generative models, Bayesian inference, and reinforcement learning. Previously he was an undergraduate student at Tsinghua University, Beijing, China.
- Volodymyr Kuleshov** is a post-doctoral scholar in the Department of Computer Science at Stanford University. His work explores applications of artificial intelligence in healthcare and personalized medicine, as well as core machine learning problems that arise in this field, touching the areas of probabilistic modeling, reasoning under uncertainty, and deep learning.
- Stefano Ermon** is an assistant professor in the Department of Computer Science at Stanford University. His research is centered on probabilistic inference, statistical modeling of data, large-scale combinatorial optimization, and robust decision making under uncertainty, and is motivated by a range of applications, in particular ones in the emerging field of computational sustainability.