Can the Mathematical Correctness of Object Configurations Affect the Accuracy of Their Perception?

Han Jiang, Zeqian Li, and Jacob Whitehill
Worcester Polytechnic Institute
hjiang@wpi.edu, zli14@wpi.edu, jrwhitehill@wpi.edu

Abstract

We investigate a new type of dataset bias based on the mathematical correctness of object configurations in visual scenes, and how this bias can affect the accuracy of computer vision models. Our experiments demonstrate how CNNs trained to detect and recognize individual objects are capable of implicitly learning simple mathematical relationships between them directly from pixel data; moreover, models that are trained with a dataset bias (e.g., all examples are mathematically correct) can suffer in performance when evaluated on test data without this bias. We found evidence for this effect in two settings: (1) object detection of math symbols in images of arithmetic expressions, and (2) object detection of moving particles from images produced by a physics simulator. Importantly, the semantic bias that we study is based not just on simple co-occurrence patterns in each image, but rather on higher-order semantic rules that generalize to unique combinations of objects not seen during training. While the magnitude of the effect was small, the accuracy difference was statistically reliable.

1. Introduction

Visual context affects object perception. Extensive research in psychology and neuroscience on human perception, as well as computer vision and machine learning research on artificially trained models, has demonstrated how the context can impact perception in both detection and recognition tasks (e.g., [4, 13, 14]). In human perceivers, the mechanisms of how surrounding objects can affect object perception include modulated visual attention as well as changes to how low-level features are integrated when forming high-level judgments about object categories [13]. Within the computer vision community, this line of research has partly motivated the collection of new datasets (e.g., CLEVR) so as to reduce biases in their ground-truth labels that could otherwise be exploited by trained models to obtain a deceptively high accuracy [10].

To date, work within machine learning and computer vision on dataset bias has focused mostly on statistical *correlations* between an object and its context that can be learned during training, e.g., if a training dataset contains boxes that are mostly red, then that statistical dependency can affect perception at test time as well. However, a related and arguably deeper question is whether a neural network's accuracy could be influenced by its understanding, or lack thereof, of *semantic* relationships between objects and their attributes that *generalize* beyond mere co-occurrence.

As a specific motivating application that we recently encountered, suppose one wishes to train an object detector to find all the math content (expressions and equations) within each frame of a collection of math tutorial videos, so that the math content in the videos can be more easily searched. Might a CNN trained on such videos learn a bias whereby the *correct* content (e.g., "$5 - 2 = 3$") is more likely to be detected as a visual object than incorrect content (e.g., "$5 - 2 = 4$")? Could such a bias be learned by generalizing the rules of arithmetic beyond the finite set of examples that were provided during training (in other words, can the machine implicitly learn that $5 - 2 = 3$ even if this specific combination of operators and operands was not part of the training set)? For another example, suppose a computer vision-based automatic homework grading system (e.g., as developed by the company GradeScope) was trained to evaluate whether each student solved a set of algebra problems correctly; would the system suffer in accuracy of detecting *individual symbols* if it was trained only on examples of *correct* solutions, in which the configurations of symbols followed the rules of algebra?

On the surface, it may seem obvious that a network trained on a dataset with some property $P = 1$ (e.g., whether all the equations rendered in the images are mathematically correct) should do better when tested on a dataset for which $P = 1$ compared to when $P = 0$ (e.g., the equations rendered in the images are often incorrect). This would be especially so if P could be learned by the model through memorization of specific objects from training images, or if P were based on simple correlations such as

"if object X appears, then object Y usually also appears". However, we argue that the answer to the question is not obvious when the bias is based on non-trivial semantic (rather than just correlational) relationships between objects (e.g., subtraction of two-digit numbers requiring "borrowing" from the 10's to the 1's place) that might be difficult for a neural network to learn even with explicit training, and when the machine must generalize to novel combinations of objects never seen during training.

In this paper, we describe a sequence of experiments to explore the influence of the mathematical correctness of object configurations in a visual scene on the accuracy of their perception by simple CNN architectures. We investigate the effect in both object recognition and object detection tasks, and in two different settings: mathematical expressions and equations rendered as images, and physical simulations of moving particles. At a high level, our results indicate that (a) neural networks are capable of learning simple mathematical relationships implicitly from how the objects appear together in images, without explicit supervision of what the objects mean or what the relationships are; and (b) the dataset bias, in terms of the mathematical correctness in the configurations of objects, can affect the network's perception accuracy at test time – even on specific configurations of objects never seen during training. Our paper contributes to the growing interest in dataset bias, as well as on causal models [16] that are valid beyond the standard "in-distribution generalization" paradigm [2].

2. Related Work

Bias in Neural Network Training: One common weak point of neural networks is that they easily overfit to biases in the training data. [1] points out that in the Visual Question Answering (VQA) dataset [3], just because a model can correctly answer some image-question pairs does not necessarily mean the model is trained well, due to the possibility of label bias in the training dataset. For example, a model might be trained to answer the question, "What covers the ground?" in a dataset in which snows always covers the ground. For the goal of helping trained models to generalize better, [10] created a new dataset (CLEVR) that minimizes the kinds of questions that do not require actual visual reasoning, thereby reducing the bias caused by the co-occurrence of two objects in the image.

In the domain of object recognition, [7] showed that CNNs trained on ImageNet [6, 17] often use textural information more than shape information. In their example, a cat with Indian elephant texture was recognized as an "Indian elephant" rather than a "cat". This kind of bias might be caused by the uniqueness of the texture of that class. In each image, the Indian elephant can have multiple shapes and poses, but almost all the textures are the same, and the texture is often easier for the network to harness for the recognition task. In [2], the authors described how non-semantic features such as color can influence the network's output. They proposed four different kinds of training regimes: in-distribution generalization, generalization under non-systematic-shift, generalization under systematic-shift and semantic anomaly detection.

Learning Mathematical Relationships: A number of works [5, 12, 18] have investigated the extent to which neural networks can be trained to solve mathematical problems. However, relatively little prior literature has explored whether neural networks can learn mathematical logic from images directly, rather than via explicit supervision. For example, [9] used two images that contained numbers as the input to a feed-forward neural networks and an image that contained the results of the two input numbers as the output. The operations could be addition, subtraction or multiplication. There was no extra information about what the characters (numbers) mean to the model. Their results showed that some mathematical concepts (addition and subtraction) could be purely learned by visual information. [11] presented a CNN based model that could learn to perform addition using input images that contained a mathematical expression, e.g., "$6 + 9$", without knowing what the characters "6" and "9" mean in advance. In [8], the authors defined a mapping from the Fashion MNIST to "0" to "9", and used this mapping to generate a new math dataset which used the Fashion MNIST examples as the numbers. The input to the model (RNNs or CNNs) were two images, and the output was also an image that contained the result of the input numbers. Their results showed that bitwise-and and bitwise-or were easier to learn than addition and subtraction. Finally, [19] found that CNN-based models also have the ability to learn some cognitive reasoning tasks such as symmetry, counting, etc. They found that, while humans can learn the tasks from just a few examples and achieve 100% accuracy after humans mastered this task, the neural networks require a large number of training examples and and cannot "master" the tasks like humans can.

3. Experiment I: Learning to Perceive Subtraction Problems

In our first experiment, we assessed whether the mathematical correctness of object configurations that are rendered as images affects the accuracy in recognizing or detecting the equations' individual objects. Here, the "objects" are symbols (0-9, -, =) that describe a mathematical equation, and the mathematical relationship between the symbols is the subtraction operation (which requires "borrowing" from the 10's to the 1's place). Because we were uncertain at the onset as to whether a CNN could implicitly learn the mathematical relationships implicitly (i.e., without supervision of the correctness) and directly from pixels, we conducted the experiment in a sequence of stages of in-

creasing difficulty.

Dataset: We considered math problems of the form $a-b=c$ and generated the set of all $n=99*(99+1)/2=4950$ unique tuples $\mathcal{T}=\{(a,b,c)\in\{0,1,\ldots,99\}^3 \mid (a>b)\wedge(c=a-b)\}$. All tuples in dataset $\mathcal{T}$ are *mathematically correct*. For instance, $\mathcal{T}$ contains the tuple $(55,23,32)$ since $55-23=32$. We then partitioned $\mathcal{T}$ into training, validation, and testing subsets ($\mathcal{T}^{\text{tr}}, \mathcal{T}^{\text{va}}, \mathcal{T}^{\text{te}}$, have 2476, 1237, and 1237 examples, respectively) such that, if (a,b,c) occurs in one subset s, then (a,c,b) must also occur in the same subset. The purpose of the latter condition was to ensure that, in order to achieve high accuracy, the network must learn the full semantics of the mathematical operation of subtraction, and not perform well just by harnessing the (relatively) simple rule that $a-b=c \implies a-c=b$. Our goal in designing this dataset and its subsets was to ensure that the network has to learn to generalize to new math problems entirely, not just novel images of previously seen same math problems.

Since we were interested in how the dataset bias of mathematical correctness can affect the perception accuracy, we thus also generated a dataset of *random* tuples $\widetilde{\mathcal{T}}^s = \{(a^{(i)},b^{(i)},c^{(\sigma(i))}) \mid (a^{(i)},b^{(i)},c^{(i)}) \in \mathcal{T}^s\}_{i=1}^n$, where $s\in\{\text{tr},\text{va},\text{te}\}$, and σ is a permutation of indices $1,\ldots,n$. Naturally, the vast majority of these will be mathematically incorrect (specifically, only 1.45%, 1.30% and 1.78% of the tuples were correct in the training, validation and testing subsets, respectively). Since the marginal probability distributions (within each of the training, validation, and testing subsets) $P(c)$ are the same in both $\mathcal{T}^s$ and $\widetilde{\mathcal{T}}^s$, the baseline accuracy of just guessing the majority class for c in the test set is also the same.

Hypotheses: A computer vision model trained on mathematically correct data can recognize the digits of c using two alternative pathways (see Figure 1): (1) perceive c from its pixels; or (2) predict c by perceiving a and b and then subtracting them. A network trained on random data, on the other hand, can only use pathway (1). Hence, we hypothesize that the following relationships about the symbol recognition accuracy (averaged over all 8 symbols in each equation) will hold:

1. *Train Random, Test Random = Train Random, Test Correct*. If the network is trained on $\widetilde{\mathcal{T}}^{\text{tr}}$, then the symbol recognition accuracy is independent of the mathematical correctness of the relationship between (a,b) and c.

2. *Train Random, Test Random > Train Correct, Test Random*. The network trained only on correct tuples will suffer when it is tested on random tuples, since pathway (2) above will usually be misleading.

3. *Train Correct, Test Correct > Train Correct, Test Random*. Same reason as hypothesis 2.

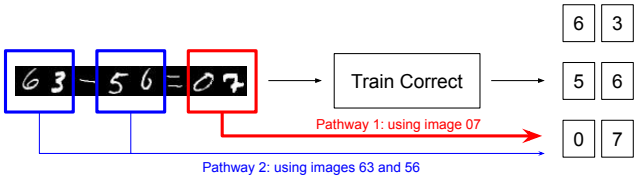


Figure 1. Two alternative pathways for how the "Train Correct" network can classify the symbols in mathematically correct expressions.

4. *Train Correct, Test Correct > Train Random, Test Correct*. The network trained only on correct tuples will benefit from being able to rely on two alternative pathways (instead of just one), especially when the images of the digits are unclear or are noisy.

3.1. Stage 1: Learning to Subtract Numbers

We first wanted to verify that a CNN that receives an image of a novel (i.e., not seen during training) two-digit subtraction problem $a-b$ can correctly compute the answer c with high accuracy. We represented each number (a or b) using two digits (e.g., a is rendered as a_1a_2) by including a leading 0 if necessary. For each digit, we randomly sampled an MNIST image of the appropriate class and concatenated them (along with an = symbol) to produce an image of the form $a_1a_2-b_1b_2=$. (See line #1 in Table 1.)

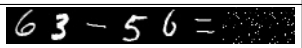
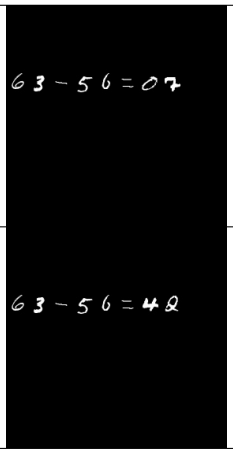
Methods: We trained a simple CNN on $\mathcal{T}^{\text{tr}}$ (and $\mathcal{T}^{\text{va}}$ for early stopping) with 4 convolutional layers followed by 2 dense layers (50 neurons each) with batch normalization and dropout using SGD (lr=5e-3). Accuracy was measured as the fraction of examples in which *both* digits of $c=c_1c_2$ were correctly predicted by the network.

Results: When tested on $\mathcal{T}^{\text{te}}$, the network achieved 90.90% accuracy. For comparison, the baseline accuracy for just guessing the majority class in the test set was 2.9%. (Note that the distribution $P(c)$ in $\mathcal{T}$ is not uniform, since there are more tuples (a,b) that yield small c than those that yield large c.) While not perfect, this network provides a proof-of-concept that a network can learn the subtraction operation with high accuracy on subtraction problems $a-b$ not seen during training.

3.2. Stage 2: Recognizing Digits in Equation Images

Next we investigated whether the mathematical correctness of the tuples (a,b,c) used to train and/or test the neural network affects the accuracy of the CNN in recognizing all the individual symbols (0-9, -, =) in images of the form $a_1a_2-b_1b_2=c_1c_2$. The networks we train have the same architecture as in Stage 1, except that the network takes an input image of size 28×224 (since there are 8 input symbols in total) and produces 8 different one-hot vectors (with

Table 1. The various datasets and tasks we used to train the networks in Experiment I on learning to perceive mathematical equations.

	Network Input		**Target**	
#	Description	Example ($63-56=07$)	Description	Example
1	Image of $a_1a_2-b_1b_2=$		One-hot codes of c_1, c_2	$[1,0,0,0,0,0,0,0,0,0]$, $[0,0,0,0,0,0,0,1,0,0]$
2	Image of $a_1a_2-b_1b_2=c_1c_2$		One-hot codes of $a_1, a_2, -, b_1, b_2, =, c_1, c_2$	$[0,0,0,0,0,0,1,0,0,0,0,0]$, $[0,0,0,1,0,0,0,0,0,0,0,0]$, ... $[0,0,0,0,0,0,0,1,0,0,0,0]$
3	Image of $a_1a_2-b_1b_2=c_1'c_2'$		One-hot codes of $a_1, a_2, -, b_1, b_2, =, c_1', c_2'$	$[0,0,0,0,0,0,1,0,0,0,0,0]$, $[0,0,0,1,0,0,0,0,0,0,0,0]$, ... $[0,0,1,0,0,0,0,0,0,0,0,0]$
4	Image of $a_1a_2-b_1b_2=$(noise)		One-hot codes of $a_1, a_2, -, b_1, b_2, =, c_1, c_2$	Same as #2
5	Image with $a_1a_2-b_1b_2=c_1c_2$ as subimage		Bounding boxes of $a_1, a_2, -, b_1, b_2, =, c_1, c_2$	(0,28,28,28), (28,28,28,28), ...
6	Image with $a_1a_2-b_1b_2=c_1'c_2'$ as subimage		Bounding boxes of $a_1, a_2, -, b_1, b_2, =, c_1', c_2'$	(28,140,28,28), (56,140,28,28), ...

12 elements each for 0-9, -, and =) as output. We conduct a 2x2 experimental design: we train the network on either mathematically correct equations or on random data; and then we test the network on either mathematically correct or on random data.

Methods: We trained the network (SGD with lr=3e-4) on *either* mathematically correct examples $\mathcal{T}^{\text{tr}}$ *or* random examples $\widetilde{\mathcal{T}}^{\text{tr}}$. (See lines #2 and #3 in Table 1; note that c_1' and c_2' refer to the two digits of a c from a *random* example in $\mathcal{T}$, as described in the beginning of Section 3.) We then tested each of the two networks on either $\mathcal{T}^{\text{te}}$ or $\widetilde{\mathcal{T}}^{\text{te}}$ and compared accuracy within the 2x2 experimental design matrix. We trained the networks using 5 random seeds and averaged the accuracy results to reduce variance.

Results: Accuracy numbers are shown in Table 2. We use t-tests (paired or unpaired, depending on the comparison) to check for statistical significance. We evaluate each of our four hypotheses below:

1. *Train Random*, *Test Random* is not stat. sig. different ($p = 0.9062$) from *Train Random*, *Test Correct*; this supports hypothesis (1).

Table 2. Mean digit recognition accuracy (std.dev.) in subtraction problem images ($a-b=c$)

	Test Correct	Test Random
Train Correct	96.65% (0.185%)	93.44% (0.177%)
Train Random	95.46% (0.192%)	95.46% (0.208%)

2. *Train Random*, *Test Random* is stat. sig. higher ($p = 5.467 \times 10^{-7}$) than *Train Correct*, *Test Random*; this supports hypothesis (2).

3. *Train Correct*, *Test Correct* is stat. sig. higher ($p = 8.017 \times 10^{-8}$) than *Train Correct*, *Test Random*; this supports hypothesis (3).

4. *Train Correct*, *Test Correct* is stat. sig. higher ($p = 1.954 \times 10^{-5}$) than *Train Random*, *Test Correct*; this supports hypothesis (4).

In sum, these results indicate that, despite imperfect learning of the subtraction operation (90.90% accuracy from Stage 1), the dataset bias of the training set in terms

of the mathematical correctness of the object configurations can still impact testing accuracy of individual symbol recognition.

3.3. Stage 3: Noisy c

In a follow-up experiment to understand better the results in Stage 2, we investigated what happens to the networks' predictions when the image of c is replaced entirely by noise. In particular, we conducted an experiment using the images rendered from $\mathcal{T}$ only (i.e., mathematically correct expressions), except that – during *testing* – the subimage corresponding to the two symbols in $c = c_1c_2$ was replaced by pure noise. (See line #4 in Table 1.) We compared two models: *Train Correct, Test Noisy* c and *Train Random, Test Noisy* c.

Methods: Same as Stage 2, except that we computed test accuracy on just the two symbols in $c = c_1c_2$.

Results: For *Train Correct*, the mean accuracy (std. dev.) was 3.06% (0.265%), whereas for *Train Random*, the accuracy was 1.85% (0.563%); the difference is stat. sig. ($p = 2.04 \times 10^{-4}$). These results further support the hypothesis that the model *Train Correct* can harness two prediction pathways. It also underlines how the dataset bias can be beneficial: if it is known a priori that the images at test time will always be mathematically correct, then the network can be made more robust (in terms of individual symbol recognition accuracy) to noise if it is trained on only correct data.

3.4. Stage 4: Detecting Digits in Larger Images

In our last experiment on perception of images of two-digit subtraction problems, we extended the task to object *detection*: Specifically, we train neural networks both to locate and to classify every symbol (0-9, - =) in the rendered image. (See lines #5 and #6 in Table 1.)

Methods: We used YOLO (v1) [15] as the object detection architecture. In contrast to the original network design, the YOLO in our experiments predicted exactly 1 symbol per grid cell. We generated images containing a random equation from $\mathcal{T}$ or $\widetilde{\mathcal{T}}$ placed onto a random location in a black 280×280-pixel background. When generating the images, each symbol was always placed in the middle of a YOLO grid cell.

We trained the networks using SGD (lr=1e-4), batch size of 100, for a maximum of 5 epochs using early stopping. Mean average precision (mAP) was used as the accuracy metric. We trained 2 instances of each network (*Train Correct*, and *Train Random*) to enable statistical significance testing.

Results: Mean average precision values are shown in Table 3. Similar to Stage 2, we find support for hypotheses (1), (3), and (4) because the differences between the corresponding pairs of cells in the table were statistically significant. However, in contrast to Stage 2, there was no statistically significant difference between *Train Random*, *Test Random* and *Train Correct*, *Test Random*.

Table 3. Digit Detection Accuracy (mAP) in subtraction problem images ($a - b = c$)

	Test Correct	Test Random
Train Correct	98.94% (0%)	98.30% (0.120%)
Train Random	98.26% (0.05%)	98.26% (0.05%)

4. Experiment II: Learning to Perceive Algebra Problems

In the next experiment we went beyond simple subtraction and explored whether mathematical correctness bias could affect individual symbol recognition accuracy in algebra problems of one variable.

Dataset: The algebra problems were all of the form $pa + q = r$, where a is the variable to be solved, each constant p, q, is an integer between -9 and $+9$ (inclusive), with the additional constraint that the solution $a = (r - q)/p$ was required to be an integer between -5 and +5 (inclusive). From each tuple (p, q, r) representing an algebra problem, we generated images containing two lines of content: The first line represented the equation, whereby the symbols in the rendered equation were randomly commuted according to standard rules of algebra (e.g., the problem $pa + q = r$ was sometimes rendered as $q + pa = r$, $r = pa + q$, or $r = q + pa$); all re-orderings of the same equation were attributed to the same algebra problem (p, q, r) and were always placed into the same data fold (train, validation, test) to avoid data leakage. The second line represented a putative solution to the algebra problem in the form $a = c$. In mathematically correct algebra problems, the value of c equals the true answer $(r-q)/p$. In random problems, c was picked uniformly at random from -5 to $+5$ (this resulted in 13.03%, 12.06% and 14.06% of the solutions being correct in the training, validation and testing subsets, respectively).

Methods: Analogously to Stage 2 of Experiment I, we trained a YOLOv1 to detect and recognize every digit of algebra problems that were placed onto larger images; see 2. We used a 2x2 experimental design, as before: { *Train Correct*, *Train Random* } $\times$ { *Test Correct*, *Test Random* }. For each of the 4 conditions, we trained 2 models for statistical significance testing.

Results: Results are shown in Table 4. In short, there was virtually no difference between conditions. It seems that the YOLOv1 detector was not able to learn the algebraic relationship between p, q, r and the solution for a to high enough accuracy so as to influence the detection accuracies.

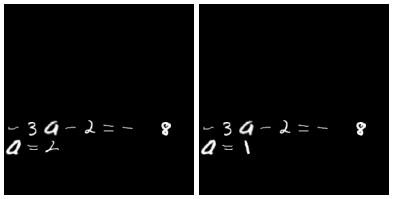


Figure 2. Examples of mathematically correct (left) and incorrect (right) algebra problems (Experiment II), where "correctness" is defined in terms of consistency between the putative solution in the second line to the problem statement in the first line.

Table 4. Detection accuracy (mAP) in the algebra problem images.

	Test Correct	Test Random
Train Correct	98.56%(0.007%)	98.56%(0.014%)
Train Random	98.58%(0.113%)	98.58%(0.113%)

5. Experiment III: Learning to Perceive Moving Particles

In our final experiment, we switch to a new task to explore whether the trends we found when perceiving subtraction problems also occur in a different setting: physics simulations of moving and colliding particles.

Dataset: We simulated the positions of two particles (one yellow, one red) at three equally spaced timesteps ($t = 0, 1, 2$sec). In the mathematically correct dataset (which we call $\mathcal{P}$), the particles (radius of 4.5 pixels, with starting position chosen uniformly at random between 4 and 45 pixels along each axis) both initially move at a constant speed (chosen uniformally at random from 4 to 12 pixels/sec) toward each other; if and when they collide within the 2 second interval, their collision conserves both momentum and kinetic energy. Each image in $\mathcal{P}$ is the concatenation of the renderings (each 50×50 pixels with 3 color channels) of the particles (plus some random background noise) at the three timesteps. Figure 3 (top) shows an example of an image in $\mathcal{P}$. In contrast, the random dataset $\widetilde{\mathcal{P}}$ contains a mixture of images, half of which are correct (drawn from $\mathcal{P}$) and half of which are incorrect (whereby the coordinates of the balls at the three timesteps are generated randomly).

Methods: We trained a YOLOv1 to detect each the 6 particles in each input image, similar to Section 3.4. Since training was slow, we trained just one neural network for each experimental condition.

Results: Table 5 shows the mean Average Precision (mAP) of the networks *Train Correct* and *Train Random* evaluated on either *Test Correct* or *Test Random*. Figure 4 shows examples of object detections. The results are consistent with all four of our hypotheses from Section 3. Figure 4 shows examples of the detections for the different conditions.

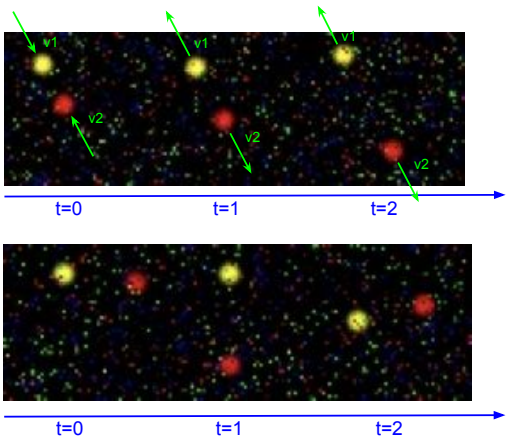


Figure 3. Examples images in in the moving particles experiment. Top: mathematically correct example, along with superimposed arrows (for the reader, not rendered in the actual dataset) showing the initial velocities of the particles. Bottom: random (and mathematically incorrect) example.

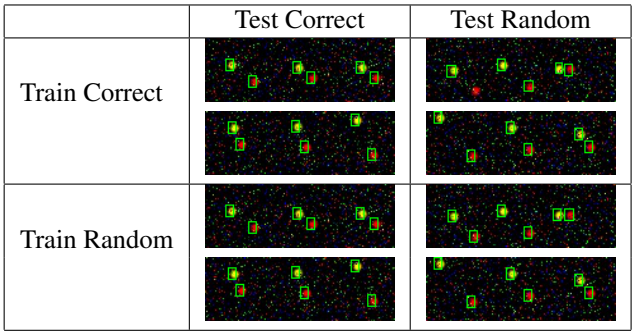


Figure 4. Examples of object detections in the moving particles experiment.

Table 5. Particle detection accuracy (mAP) in the Colliding Particles experiment.

	Test Correct	Test Random
Train Correct	99.01%	80.74%
Train Random	98.88%	98.89%

6. Conclusions

We have conducted object recognition and object detection experiments, on images of arithmetic (subtractive) expressions, algebra problems, and colliding particle simulations, to explore whether dataset bias regarding the mathematical correctness/incorrectness of the object configurations can impact the accuracy of the objects' perception. For the subtraction problems and particle simulations, we found that the neural networks were, with enough training data, capable of implicitly learning the semantic relation-

ship and generalizing to new scenes containing instances of the relationship that never were seen during training; moreover, the implicitly learned semantic rules yielded small but reliable accuracy differences when tested on a dataset with a different semantic bias. On the algebra problem task, no such effect was observed, possibly because the semantic relationship was too challenging for the network to infer implicitly and directly from pixels; it is however conceivable that more powerful recognition architectures might still be able to learn the relationships.

Importantly, our results go beyond mere correlational label bias and instead address the semantic question of whether high-level relationships can be generalized and influence the network's perceptual accuracy. To our knowledge, these results are the first to demonstrate how mathematical relationships between objects can be learned and influence perception accuracy.

On one hand, our results suggest that, if it is known ahead of time that *all* data at test time will adhere to certain mathematical constraints, then it is worth optimizing the network on exactly the same constraints at training time, as this may yield an accuracy advantage. On the other hand, if the mathematical correctness of the objects' configurations can vary from both correct to incorrect, then it may be useful to train the network accordingly. Particular for educational applications on automatic math problem grading, which served as a concrete motivation for this paper, it may be important to assure the students, teachers, and parents, that the recognition accuracy of the *individual symbols* in students' submissions is the same for everyone, regardless of whether their overall math solution was right or wrong.

Acknowledgements: This research was supported by the NSF National AI Institute for Student-AI Teaming (iSAT) under grant DRL 2019805. The opinions expressed are those of the authors and do not represent views of the NSF. This research was also supported by ONR grant N00014-18-1-2768, and by NSF CAREER grant 2046505.

References

[1] Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356*, 2016. 2

[2] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. Systematic generalisation with group invari-ant predictions. 2021. 2

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International conference on computer vision*. 2

[4] Christel Bidet-Ildei, Manuel Gimenes, Lucette Toussaint, Yves Almecija, and Arnaud Badets. Sentence plausibility influences the link between action words and the perception of biological human movements. *Psychological research*, 81(4):806–813, 2017. 1

[5] Sungjae Cho, Jaeseo Lim, Chris Hickey, and Byoung-Tak Zhang. Problem difficulty in arithmetic cognition: Humans and connectionist models. 2019. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 2

[8] Qian Guo, Yuhua Qian, and Xinyan Liang. Mining logic patterns from visual data. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 620–627. IEEE Computer Society, 2019. 2

[9] Yedid Hoshen and Shmuel Peleg. Visual learning of arithmetic operation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. 2

[10] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 1, 2

[11] Shuaicheng Liu, Zehao Zhang, Kai Song, and Bing Zeng. Arithmetic addition of two integers by deep image classification networks: experiments to quantify their autonomous reasoning ability. *arXiv preprint arXiv:1912.04518*, 2019. 2

[12] Bastien Nollet, Mathieu Lefort, and Frédéric Armetta. Learning arithmetic operations with a multistep deep learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 2

[13] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 1

[14] Keith Rayner, Tessa Warren, Barbara J Juhasz, and Simon P Liversedge. The effect of plausibility on eye movements in reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(6):1290, 2004. 1

[15] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 5

[16] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. 2

[17] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020. 2

[18] Artit Wangperawong. Attending to mathematical language with transformers. *arXiv preprint arXiv:1812.02825*, 2018. 2

[19] Zhennan Yan and Xiang Sean Zhou. How intelligent are convolutional neural networks? *arXiv preprint arXiv:1709.06126*, 2017. 2