How Does Visualization Help People Learn Deep Learning? Evaluating GAN Lab with Observational Study and Log Analysis

Minsuk Kahng*
Oregon State University

Duen Horng (Polo) Chau†
Georgia Institute of Technology

Abstract

While a rapidly growing number of people want to learn artificial intelligence (AI) and deep learning, the increasing complexity of such models poses significant learning barriers. Recently, interactive visualizations, such as TensorFlow Playground and GAN Lab, have demonstrated success in lowering these barriers. However, there has been little work in evaluating these tools with human subjects. This paper presents two studies on evaluating GAN Lab, an interactive tool designed to help people learn how Generated Adversarial Networks (GANs) work. First, through an observational study, we investigate how the tool is used and what users learn from their usage. Second, we conduct a log analysis of the deployed tool to investigate how its visitors engage with GAN Lab. Based on the studies and our experience in developing and successfully deploying the tool, we provide design considerations and discuss further evaluation challenges for interactive educational tools for deep learning.

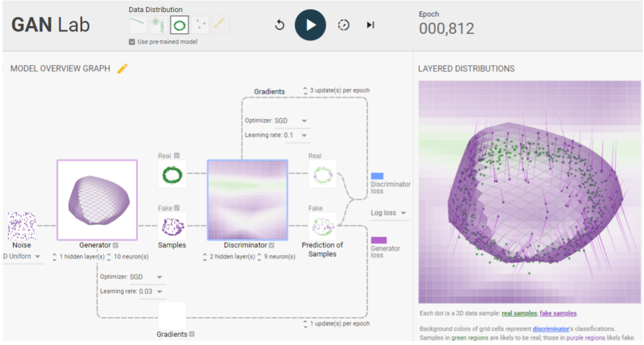
Index Terms: Human-centered computing—Visualization—Visualization design and evaluation methods

1 Introduction

With the recent advances in artificial intelligence (AI) and deep learning, a rapidly growing number of people want to learn a variety of new deep learning models. However, the increasing complexity of such models poses significant learning barriers. Recently, interactive visualizations have demonstrated success in tackling this challenge [6, 9, 10, 16, 20, 23]. For instance, TensorFlow Playground [20] allows users to directly manipulate a visualization of neural networks, which has been used to educate employees at Google about deep learning [1]. Furthermore, an increasing number of explorable tools, often called *explorable explanations*, have been developed [6, 17, 22], and a series of the VISxAI workshop at IEEE VIS has successfully featured these tools over the past couple years.[1]

While these interactive educational tools have gained popularity and research interest, there has been little work in formally evaluating them. Only few works have been published as academic papers [9, 16, 20, 23], and some of which include usage scenarios [9]. Evaluation of this new type of tools that focus on educational aspects could be different from that for typical visual analytics tools designed for interpreting the inner-workings of models [6] or that for interactively building models [2].

This paper presents a follow-up study of our IEEE VAST'18 paper (TVCG track) on GAN Lab [9], an open-source interactive educational tool for Generated Adversarial Networks (GANs), a popular but difficult-to-understand deep learning models. GAN Lab is the first tool designed to help people learn and experiment with complex GAN models in web browsers. In this paper, we are primarily interested in evaluating GAN Lab with the following two research questions: (1) what features in GAN Lab do users use most and what do they learn from their usage? (2) how do visitors of our deployed website engage with the tool? To answer these questions, we conducted two studies. First we conducted a small observational study to investigate how our target users would use GAN Lab. Next, to investigate whether users of the deployed tool actively engaged with GAN Lab by using a variety of features it provides, we collected and analyzed 20 weeks of interaction log data from anonymous users (total 124,293 click events by 6,828 users).



3 update(s) per epoch

Optimizer: SGD

Learning rate: 0.1

D Uniform

1 hidden layer(s) 10 neuron(s)

2 hidden layer(s) 9 neuron(s)

Optimizer: SGD

Learning rate: 0.03

1 update(s) per epoch

Each dot is a 2D data sample: real samples; fake samples.

Background colors of grid cells represent discriminator's classifications. Samples in green regions are likely to be real; those in purple regions likely fake.

Figure 1: GAN Lab visualizes the structure of GAN models and allows users to interactively train and experiment with the models, helping them actively and playfully learn about GANs.

2 Background: GAN Lab for Learning Deep Learning in Browser

GAN Lab [9], an interactive visualization tool for learning GANs, was designed and developed through a collaboration between Georgia Tech and Google's People+AI Research (PAIR) group[2]; the authors of this paper were part of the team. GAN Lab supports a growing population of people who want to learn deep learning, but had a hard time doing so because of the complexity of modern deep learning models. GANs [4] is a great example of such models. To lower the learning barriers, we built GAN Lab and presented a paper at VAST'18 in the TVCG track [9].

GAN Lab (Fig. 1) enables users to interactively train a GAN, tweak its hyperparameters, and study how it responds to generate data distributions. GAN Lab's visualization techniques work in tandem to help people understand complex GAN concepts. The interface tightly integrates a *model overview graph* that summarizes GAN's structure as a graph (Fig. 1 left), selectively visualizing components crucial to the training process; and a *layered distributions view* (Fig. 1 right) that helps users interpret the interplay between the *generator* and *discriminator*, the two key components of GANs.

GAN Lab was developed as a standalone browser-based tool, overcoming significant barriers to learning deep learning. Conventional deep learning visualization systems often require servers that use significant computing resources and expensive hardware, like GPU, which may not be accessible to our target users. We overcame this challenge by putting everything, including model training and

*e-mail: minsuk.kahng@oregonstate.edu
†e-mail: polo@gatech.edu

[1] https://visxai.io/

[2] https://pair.withgoogle.com/

visualization, into browsers, by using TensorFlow.js [21], a deep learning library written in JavaScript. Thus, people can access GAN Lab using only their web browsers without the need for specialized backend, which significantly broadens people's access to GAN lab.

Deployment. GAN Lab was open-sourced and launched in September 2018 at `https://poloclub.github.io/ganlab/`. GAN Lab was a great success. Our release went viral and received significant attention. Within the first year of release, more than 70,000 people from over 160 countries tried it out. It has also been used to teach the concept of GANs in classes, including Georgia Tech's graduate-level deep learning course by Dr. Batra.

3 Observational Study

To investigate how GAN Lab's target users (e.g., students aspired to learn about GANs) would use the tool and learn about the models, we first conducted a small observational study. This section describes our study design and findings.

3.1 Experiment Design

Participants. Six participants were recruited through our institution's mailing list for those who are interested in machine learning. We pre-screened participants to ensure that they have at least basic knowledge of deep learning and GANs (e.g., taken a deep learning course or at least heard of GANs). Five participants were Ph.D. students who had taken a deep learning course, and one was an undergraduate student who had research experience. They self-reported their level of knowledge on deep learning, with an average score of 3.3 on a scale of 0 to 5 (0 being "no knowledge" and 5 being "expert"); and that on GANs with an average score of 2.5 (on the same scale). No participant has used or heard about GAN Lab before.

Procedure. The study was conducted through BlueJeans video conferencing. After the participants signed their consent forms electronically, they were provided a 5-minute overview of GANs, followed by a 5-minute tutorial of GAN Lab, which described its visualizations and features. After that, the participants freely explored using GAN Lab on their computer's web browser. They were asked to think aloud and share their computer screen with us during the study. They could ask for questions when necessary. After they used the tool, the participants were asked to fill out questionnaires. The study took about 50 minutes, and each participant was compensated with an Amazon $15 gift card for their time.

3.2 Questionnaire Results

Subjective ratings. We measured several aspects of GAN Lab using 7-point Likert scales (7 being *Strongly Agreed*; 1 being *Strongly Disagreed*). Table 1 shows the average ratings for the 12 questions we asked. The participants found that GAN Lab was easy to learn, easy to use, helpful to understand several aspects of GANs, and likeable overall. Specifically, all six participants found GAN Lab easy to learn to use (i.e., rated 6 or 7), and all but one participant agreed that GAN Lab was easy to use, they enjoyed using it, and they would like to use software like to learn machine learning. Five questions starting with "helpful to understand" are asking whether GAN Lab improves their understanding of certain aspects of GANs. The question that received the highest average rating was on what a GAN model is composed of, which indicates that GAN Lab's visualization was effective. In addition, the only question that all participants agreed was on the understanding of the effects of hyperparameters, related to the GAN Lab's interactive experimentation features. Even with a variety of features that aim to improve the understanding of the generator (which is one of the most difficult parts of GANs), participants reported that it was relatively harder to understand the generator, in terms of the average rating (i.e., 5.5), while the value is high enough to say it is positive.

Qualitative feedback. We asked participants for feedback on GAN Lab. Participants liked a variety of visualizations and features it provided. For example, multiple participants said they liked GAN Lab's visualizations that evolve as the training process progresses. One participant said "*I liked the updated visualizations of the manifold, gradients, etc. I liked these because it provided insight as to how the GAN was evolving in time, which provides insight into how it works and what the end goal of a GAN is.*" Another said "*I did learn more properly how the GANs actually evolve, as I did not fully understand how they operated before. I don't think my DL [(deep learning)] professor explained as nicely as how this tool demonstrated.* " In addition, multiple participants particularly liked the feature for adjusting hyperparameters. We report them and other feedback more in detail in the next subsection.

Table 1: Subjective ratings about GAN Lab using 7-point Likert scales (7: *Strongly Agreed*. 1: *Strongly Disagreed*).

Question	**Avg.**
Easy to learn how to use	6.3
Easy to use	6.3
Helpful to understand what constitutes a GAN model	6.5
Helpful to understand the training process of GANs	6.0
Helpful to understand what the generator is doing	5.5
Helpful to understand what the discriminator is doing	6.2
Helpful to understand how hyperparameters affect results	6.2
Helpful to get new insight about GANs	5.8
I felt confident when using the tool	5.8
It improves the effectiveness of my learning	5.7
I enjoyed using GAN Lab	6.5
I would like to use software like GAN Lab to learn ML	6.5

3.3 Key Findings

Below we summarize key findings from our observations and qualitative feedback from the participants.

Rapid hypothesis testing. Among the features of GAN Lab, many participants particularly liked the one for dynamically adjusting hyperparameters while a model was being trained. This feature enabled them to form hypotheses based on prior experience in machine learning and rapidly test them using GAN Lab. For example, one participant increased the learning rate (using its drop-down menu) to test if it helps speed up the training. Another participant said "*I really liked the features of the hyperparameter tuning [...], and learning all the different hyperparameters that can affect them are making me think of different ways to optimize GANs.*" This capability for rapid hypothesis testing in GAN Lab is not possible in conventional deep learning workflows because they often require retraining the model each time a user adjusts a hyperparameter.

Building intuition through dynamic experiments. The ability to adjust hyperparameters in GAN Lab also helps users build intuition about the behaviors induced by the model's training process. One important characteristic of GANs is the dynamic interplay between the two components: generators and discriminators. A participant said "*[the] ability to change training parameters such as number of updates on the fly was nice. It really helps you build intuition to see how the discriminator and generator interact.*" One usage pattern participants particularly liked was updating either the generator or discriminator while disabling the update of the other. By default, the training process alternates between the generator and discriminator (in each iteration), so it can be hard for novices to understand their individual contribution to the training progress. By disabling one of them, users can more easily observe how each component works and how the model reaches an equilibrium that balances the two components.

Validating knowledge from literature. Participants who are familiar with the literature of deep learning and GANs found GAN Lab useful for validating knowledge they acquired from research articles. For example, one participant remembered that GANs would often encounter the problem called *mode collapse*, especially when a distribution contained disjoint modes [11]. This participant was interested in reproducing this phenomenon by training a model with such a distribution. He also wanted to use a different loss function that might mitigate this issue, as suggested in the literature. This observation suggests that interactive tools like GAN Lab may help not only novices learn the basic concepts of models, but also researchers and practitioners validate knowledge they learned from the literature, which could help them build trust in the model's training process.

Beginners need further guidance. We observed that participants less familiar with GANs needed more guidance to help them fully enjoy the tool. Some were not sure about what to try. One said "*helpful to [provide descriptions] of what GANs training scheme "works" and what "doesn't work."*" Although we wanted users to self-discover relationships between hyperparameters and results by actively playing with the tool, it might be beneficial for us to also provide step-by-step exercises that would guide users' experimentation, similar to how TensorFlow Playground has been integrated into Google's machine learning course material on the web [1]. The course includes a series of exercises which learners can follow. For example, in the chapter on learning rates, learners are asked to try different learning rates and compare the results.

4 Log Analysis

As mentioned earlier, we have deployed GAN Lab on the web, where anyone can visit and play with GAN Lab using their browsers. Since launched in September 2018, it has received significant attention. More than 130,000 people from over 180 countries tried it out as of July 12, 2020 (according to Google Analytics).

While we were excited about this number of visitors, we were curious if they actually engage with GAN Lab and were able to use a variety of its features. Oftentimes when we, as a visualization designer, develop a new visualization tool for domain experts, we closely work with target users and provide detailed tutorials to them, however, GAN Lab has been deployed for the public, and anyone can access to our website and use it without any direct one-on-one guidance from us. Although the website contains a short introduction of GANs and descriptions of the tool, we were a little worried if they were able to find several features in GAN Lab and play with them.

Thus, to investigate how users of the deployed tool engage with GAN Lab by using a variety of features it provides, we conducted a study on an analysis of users' interaction log. Analyses of users' interaction histories have been widely used to evaluate visual analytics tools [3, 5, 18]. Previous studies demonstrated that careful examinations of a user's interaction log can recover the user's reasoning process [3]. Both automated techniques and manual reviewing have been used [5]. We used a semi-automated approach which we first manually identified common *actions* and automatically extracted the identified actions from the logs.

4.1 Data Collection

We have collected anonymous interaction logs from the deployed website. The logs mainly include users' clicks on HTML elements and their scrolling events. We analyzed 20 weeks of data (140 days) collected from July 27 to December 13, 2019. We did not collect data from users located in European countries determined based on their computers' timezone information because of IRB-related issues, and we also did not use data for users who opted out by clicking the corresponding link on the website. The study has been approved by Georgia Tech's IRB, and consent forms were waived. The collected data are stored in databases on Google Cloud.

Summary statistics. The collected data include 124,293 click events by 6,828 users (18.2 clicks on average). They do not include users who left the website without any clicks. Among 6,828 users, 2,813 users clicked the elements on GAN Lab at least 10 times, 1,239 users at least 25 times, 512 users at least 50 times, and 183 users even more than 100 clicks. To analyze the behavior of users who had sufficiently interacted with the tool, we decided to analyze the interaction logs for the 1,239 users who clicked the elements at least 25 times. An average click count by these 1,239 users is 69.0.

There are 91 different HTML elements clicked by at least one of the all 6,828 users. The elements range from the play button to several drop-down menus. For example, the most popular element was the play button located on the top of the interface, which was clicked at least one time by 1,224 users among the 1,239 users.

4.2 Method for Identifying Common Actions

While log analysis at the level of HTML elements provides a basic information of usage statistics, we are interested in higher-level semantically meaningful behaviors of users. Thus, we decided to identify a list of common *actions*, similar to Gotz and Zhou [5]'s *action* tier, a richer level of semantics not found in lowest-level user interaction event (e.g., mouse click).

To identify common actions, we iteratively transform a set of GAN Lab's features into actions based on their sequential usage patterns with the help of an interaction timeline visualization. We first build a hierarchy of features for GAN Lab and categorize the 91 HTML elements based on the hierarchy. To explore the log data in terms of the hierarchy, we developed a timeline visualization that shows each user's click sequence as a column, similar to that used in the literature [3]. Each clicked HTML element is shown with a colored category label, so that we can visually group sequence subsets and recognize patterns. The user columns can also be filtered based on whether a user clicked a particular element, to help us explore a large number of user sequences. After exploring this visualization, we transform the features into an initial set of actions. Then we iteratively revise the list by further exploring the visualization, and when the action list is updated, the visualization is also updated. For instance, a top-level feature (e.g., hyperparameter tuning) is transformed into two different actions (e.g., one for changing the size of models before training and the other for adjusting optimization parameters, such as learning rates, during training) because there exist two distinct patterns for the feature.

Once we identify a list of the common actions, we have written a script that automatically finds matching patterns from the logs. For example, to determine whether a user had adjusted hyperparameters while a model was being trained, the script first selects users who clicked corresponding HTML elements (e.g., item in the dropdown menu for learning rates) and checks if the iteration count had been increased after the click event. We have iteratively refined the script by incrementally adding constraints, to accurately reflect the identified actions. For instance, to determine if a user had used the slow-motion mode, we first simply checked if they clicked the button for the slow-motion mode, however, we soon realized that some users clicked the same button right after their first click, which means they unlikely used the feature, so we have revised the script to count users only when they used the feature at least for 10 seconds.

4.3 Results

Table 2 shows the list of 9 common actions sorted by the number of users (among the 1,239 users) who performed each of them. For example, the fourth row indicates that 697 users (56% out of 1,239) drew at least one data distribution by themselves using the GAN Lab's feature for drawing new distributions and trained a model for it. The results demonstrate that many of users were able to play with GAN Lab by using a variety of features, even though all these users are anonymous users who visited our website voluntarily. For instance, a large number of users trained GAN models by selecting multiple different data distributions available on the interface (i.e., #1, #4). In addition, many users investigated the interplay between the two submodels, the generator and discriminator, by adjusting a

Table 2: List of 9 common actions sorted by the number of users who performed each of the actions at least one time (for 1,239 users who clicked any HTML elements at least 25 times)

#	Action	# of Users
1.	Select another pre-defined data distribution and train a model	1094 (88%)
2.	Read instructions on the deployed website	807 (65%)
3.	Enable and inspect the animated visualization of the generator's manifold	759 (61%)
4.	Draw a user-created distribution and train a model	697 (56%)
5.	Change the size of submodels (e.g., number of neurons for a generator or discriminator)	394 (32%)
6.	Adjust hyperparameters (e.g., learning rates) while a model is being trained	336 (27%)
7.	Enable and inspect the training process using the slow-motion mode	328 (26%)
8.	Train either a generator or discriminator while freezing the other	271 (22%)
9.	Adjust the number of training iterations for submodels (e.g., updating a discriminator three times more than a generator)	192 (15%)

parameter for one of them (e.g., train either a generator or discriminator in #8). Furthermore, many visitors directly manipulated a range of hyperparameters (i.e., #5, #6, #9).

4.4 Design Considerations for Future Deep Learning Education Tools

Based on the action-level analysis and our exploration of the log data using the visualization, we distilled three design considerations for the development of future deep learning educational tools.

Fast model training. While exploring the logs, we witnessed that some users left the website while models are being trained, especially for those with many layers or neurons that require more time to be trained. It is well known that users do not wait for computer systems if they take too much time, especially when interfaces do not provide feedback [12]. While participants in the observational study could not easily leave the study, users of deployed websites can do it. Thus, it is important for tool designers to take every effort to reduce time for model training and provide visual feedback while users are waiting. For example, we used pretrained models and generated visual feedback but not too frequently to reduce computation time [9]. It might also be possible to help users effectively use their waiting time. For instance, the log data indicated that some users read instructions and used other features while waiting.

Supporting user-provided datasets. As shown in Table 2, a majority of users selected multiple different data distributions and created new data to work with GAN Lab, while a smaller number of users played with hyperparameters for models. This might be because data are more familiar to novice users than concepts for specific ML models. Also, it might be easier for users to visually compare the model behaviors for different datasets than for different hyperparameters. For instance, some participants from the observational study created a distribution that is slightly different from a dataset provided by the tool, and they were able to clearly see how models learn differently. Thus, we suggest designers provide users with an ability to work with user-provided datasets. It would be even better if a tool enables users to play with real-world data.

Balancing between step-by-step guidance and free-form exploration. From the log data, we found several different styles of using GAN Lab. For example, some users spent time reading instructions first before they start playing with the visualization, while some others directly dived into the visualization first and scrolled down to the instructions whenever needed. As this type of deployed tools can be used by people with different expertise and learning styles, we need to take those into consideration. In Sect. 3.3, we discussed that beginners may need further guidance, possibly through step-by-step exercises [1]. We think that step-by-step exercises or Distill-style interactive articles [17] could be useful for beginners or any users who use visualizations for their first time, and interactive visualizations that users can freely explore could be more useful once they familiarize themselves with the visualizations. Future work may study on the effect of different interface styles for users with different expertise and learning styles.

5 Discussion: Measuring Understanding Level

Our observational study and usage log analysis are an early step in understanding how people may learn deep learning through interactive education tools. There remain challenges in designing studies to further evaluate the educational effectiveness of tools like GAN Lab. One important challenge is how we can design quantitative lab studies and what would be their dependent variables that measure a user's level of the understanding in machine learning (ML) models, like the use of task completion time in evaluating information exploration tools. We briefly discuss this challenge in this section.

Studies conducted in computer science education research and those for evaluating algorithm visualizations (in early 2000s) typically included pre- and post-study tests that sought to measure participants' conceptual or procedural knowledge (e.g., what is the algorithm's time complexity, what would be the next state after '17' is inserted to a binary search tree) [7]. However, test questions suitable for simpler, deterministic algorithms may not generalize to modern ML models that are often complex and probabilistic.

Thus, it would be a valuable effort to develop new ways to evaluate the educational effectiveness of interactive tools for ML. Below we present a few ideas. First, the computer science education literature has developed several methods, such as analyzing mental models or measuring self-efficacy [13, 19], and we can draw inspirations from them. Next, inspired by how visual analytics tools are evaluated [15], studies may be designed to analyze if participants discovered new insights on ML models. In addition, since the primary goal of ML learners is often in developing models for real data, it could be helpful to design studies that assess if users are able to implement actual models (e.g., a GAN model for generating new artistic images) with high accuracy.

Lastly, we wanted to note that the level of understanding is not the only dependent variable in evaluating educational tools. Another important factor to measure is the learners' engagement level [14]. A high level of engagement (e.g., spending more time and efforts) often indicates that users enjoy the tool and may likely learn more through the usage. Our log analysis provides an initial step to study how people engage with tools like GAN Lab, and we hope future studies will further investigate how the tools facilitate user engagement.

6 Conclusion

This paper presents a follow-up evaluation study of GAN Lab, a popular interactive tool for learning a popular, complex deep learning model. From an observational study, we found that GAN Lab helps users learn about GANs through multiple interactive features, such as dynamic adjustment of hyperparameters. By collecting large-scale anonymous usage logs from visitors of our deployed website, we were able to see that they used a variety of features provided by GAN Lab. We believe tools like GAN Lab have a huge potential for promoting people's understanding of AI, and hope our work will inspire more research, development, and evaluation of such tools.

Acknowledgments

We thank Martin Wattenberg and Fernanda Viégas for their advice. We also thank for feedback from the presentation of our observational study part at the EVIVA-ML workshop at IEEE VIS'19 [8]. Minsuk Kahng was partly supported by a Google PhD Fellowship.

References

[1] Neural networks: Playground exercises, machine learning crash course. `https://developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/playground-exercises`, 2019. Accessed: 12 July 2020.

[2] N. Boukhelifa, A. Bezerianos, and E. Lutton. Evaluation of interactive machine learning systems. In *Human and Machine Learning*. Springer, 2018.

[3] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning processes from user interactions. *IEEE Computer Graphics and Applications*, 29(3):52–61, 2009.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[5] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, 2009.

[6] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics*, 25(8), 2019.

[7] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing*, 13(3), 2002.

[8] M. Kahng and D. H. Chau. How does visualization help people learn deep learning? Evaluation of GAN Lab. In *IEEE VIS Workshop on EValuation of Interactive VisuAl Machine Learning systems*, 2019.

[9] M. Kahng, N. Thorat, D. H. Chau, F. B. Viégas, and M. Wattenberg. GAN Lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1), 2019.

[10] M. Li, Z. Zhao, and C. Scheidegger. Visualizing neural networks with the grand tour. *Distill*, 2020.

[11] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. In *5th International Conference on Learning Representations (ICLR)*, 2017.

[12] R. B. Miller. Response time in man-computer conversational transactions. In *Proceedings of the Fall Joint Computer Conference, Part I*, pp. 267–277, 1968.

[13] T. Naps, S. Cooper, B. Koldehofe, C. Leska, G. Rößling, W. Dann, A. Korhonen, L. Malmi, J. Rantakokko, R. J. Ross, J. Anderson, R. Fleischer, M. Kuittinen, and M. McNally. Evaluating the educational impact of visualization. *ACM SIGCSE Bulletin*, 35(4), 2003.

[14] T. L. Naps, G. Rößling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. A. Velázquez-Iturbide. Exploring the role of visualization and engagement in computer science education. *ACM SIGCSE Bulletin*, 35(2), 2003.

[15] C. North. Toward measuring visualization insight. *IEEE Computer Graphics and Applications*, 26(3), 2006.

[16] A. P. Norton and Y. Qi. Adversarial-Playground: A visualization suite showing how adversarial examples fool deep learning. In *IEEE Symposium on Visualization for Cyber Security (VizSec)*, 2017.

[17] C. Olah and S. Carter. Research debt. *Distill*, 2017.

[18] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing provenance in visualization and data analysis: an organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1), 2015.

[19] V. Ramalingam, D. LaBelle, and S. Wiedenbeck. Self-efficacy and mental models in learning to program. *ACM SIGCSE Bulletin*, 36(3), 2004.

[20] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg. Direct-manipulation visualization of deep networks. In *ICML Workshop on Visualization for Deep Learning*, 2016.

[21] D. Smilkov, N. Thorat, Y. Assogba, A. Yuan, N. Kreeger, P. Yu, K. Zhang, S. Cai, E. Nielsen, D. Soergel, S. Bileschi, M. Terry, C. Nicholson, S. N. Gupta, S. Sirajuddin, D. Sculley, R. Monga, G. Corrado, F. B. Viégas, and M. Wattenberg. TensorFlow.js: Machine learning for the web and beyond. In *Proceedings of the 2nd SysML Conference*, 2019.

[22] B. Victor. Explorable explanations. `http://worrydream.com/#!/ExplorableExplanations`, 2011. Accessed: 12 July 2020.

[23] Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, and D. H. Chau. CNN 101: Interactive visual learning for convolutional neural networks. In *CHI Extended Abstracts*, 2020.