# Research Statement, David Bau

My research focuses on Computer Vision and Explainable AI.

I believe it is critical for programmers to understand their programs. As computer science enters a new era of self-programmed systems, my research mission is to keep humans in control while our field increasingly utilizes powerful deep networks.

The key tool that enables my mission is the emergence of modularity in deep networks. Understanding the reusable modules of any program allows a programmer to debug, modify, and restructure the computation. With deep networks, building a mastery of emergent modules means understanding the network's internal organization well enough to directly reconfigure and alter its behavior. It means enabling a new type of software engineering, where human programmers can intelligently collaborate with trained systems, with both humans and systems teaching and learning concepts from each other.

Modularity is no accident; reusable concepts are the hallmark of human thought. For instance, consider the very first drawings of a horse by a 4-year old child, Heidi [1] (Figure 1a). Although the number of legs is inaccurate, the simple forms illustrate Heidi's understanding that a horse has a head, body and legs. They reveal a human impulse to see modular parts that comes even before accurate counting.

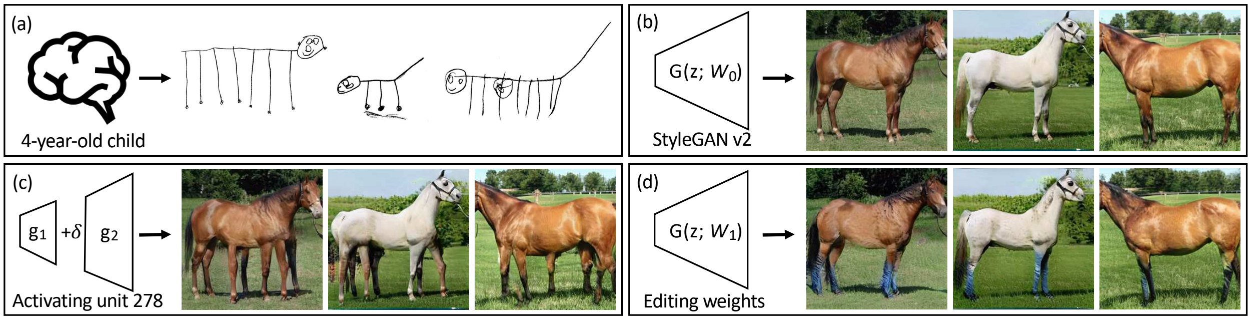

Figure 1: Comparing Heidi's horses [1] to a deep network. (a) Four-year-old Heidi's first horses reveal a modular understanding of legs and other parts. (b) Realistic horses generated by a StyleGANv2 [2] lead us to ask how the GAN represents legs. (c) Activating one unit inside the network adds legs to horses [3]. (d) Making a single rank-one change to the weights of the model adds blue jeans to standing horse legs [4, 5].

Is a deep network modular? Does it know about horse parts? Figure 1b shows the output of a current state-of-the generative network [2], trained to draw horses. Clearly this network can draw horses with standard legs, but unlike Heidi, it provides no obvious capability to add extra legs or change their style.

An understanding of the emergent modular structure of the network lets us answer this question. We investigated one way to control units in **GAN Paint** [3, 6] (ICLR 2019, http://gandissect.csail.mit.edu). In Figure 1c, a user has used our method like a paintbrush to manipulate the network's neurons to make seven-legged horses. The method identifies and activates a single unit in the network that triggers legs in the output, each leg complete with muscles, knee and hoof.

The legs can also be redefined: My work on **Model Rewriting** [4] (ECCV 2020 oral, http://rewriting.csail.mit.edu, Figure 1d) enables a user to modify the network so horses wear blue jeans. Model Rewriting treats network layers as Optimal Linear Associative Memory banks [7, 8] that can be addressed and overwritten. The method lets a user change an individual memorized rule in the network, reprogramming weights to change the appearance of legs in new images, while leaving other parts unchanged.

My research develops principles, methods, and user interfaces — such as GAN Paint, Model Rewriting, and Network Dissection [9, 10] (PNAS 2020, CVPR 2017 oral, http://dissect.csail.mit.edu) — that enable users to directly control and reprogram deep networks by exploiting *emergent modularity within learned algorithms.*

## Dissecting Deep Neworks

Should we impose modules on a network in advance? For instance, if we need a system that classifies photos of baseball fields and other sports, we might train specific subnetworks for recognizing baseballs or bats. However, dividing the problem into pre-assigned roles ahead of time assumes that the problem is solved on a conceptual level before the data is modeled. In contrast, allowing the network to learn its own modular organization lets it discover structure in the data that we do not yet know. With the right tools, the network can teach us new ideas.

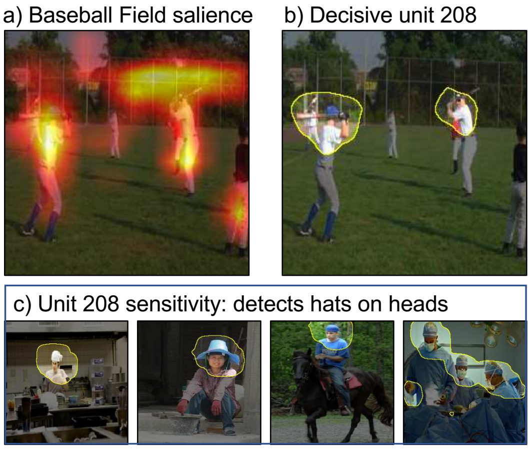
a) Baseball Field salience
b) Decisive unit 208
c) Unit 208 sensitivity: detects hats on heads

Figure 2: (a) The attention of a network can be decomposed into units (Grad-CAM [11] on a VGG scene classifier [12, 13] , classifying a baseball field). (b) The prediction for this image is insensitive to removal of most neurons in a layer conv5_3, but three units are decisive. The activation of one of the decisive units is shown. (c) Network Dissection [9] quantifies the semantic content of units by testing each unit against thousands of labeled examples. Unit 208 detects people wearing hats, explaining, in part, how this network classifies baseball fields.

For example, Figure 2 shows the use of network dissection [9, 10] to understand the operation of an ordinary deep network [12] as it classifies a baseball field. One of the three units critical to the task is neuron 208, which detects people wearing hats. The network's use of a hat detector for baseball is a helpful insight that might not be immediately obvious: baseball caps are a better pattern to seek out than balls or bats, because caps are ubiquitous in baseball while rare in other sports.

To build a global picture of the useful signals discovered by a network, I introduced a method to use a supervised model as a dissection tool. Figure 3 (Bau et al, PNAS 2020 [10]) shows emergent visual concepts in all the layers of a classifier and a generator. Each concept is identified by comparing each unit's response to the predictions of a segmenter trained (supervised) to detect 1826 human-labeled objects, parts, colors, and textures. The analysis reveals a rich vocabulary of human-understandable emergent concept units in the last layers of a classifier and the middle layers of a generator.

The inverse problem also yields insights: in ICCV 2019 (oral) [14], I quantify and visualize semantic blind spots, and show how a GAN can omit whole classes of objects after learning to fool an adversary.

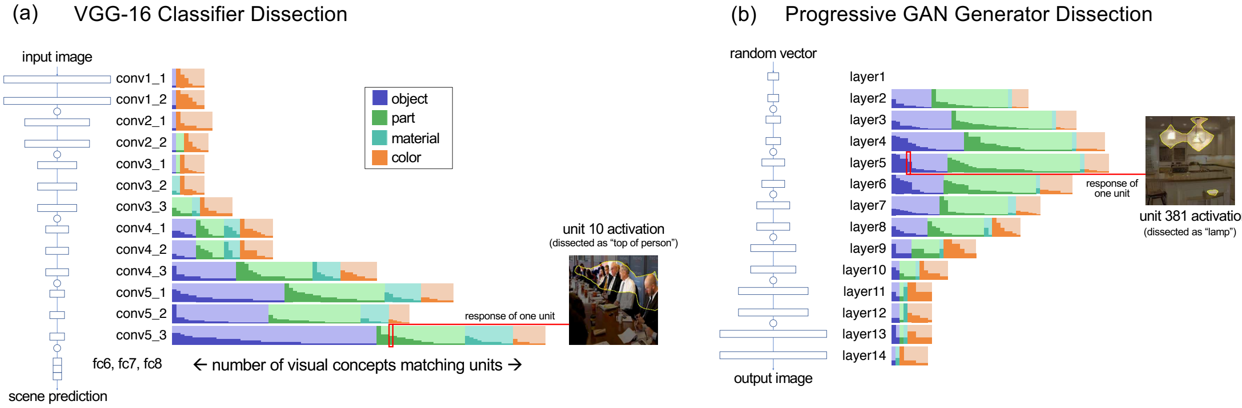

Figure 3: Using network dissection to compare emergent visual concepts in a classifier and a generator (PNAS 2020 [10]). (a) Dissecting emergent concepts in the layers of a VGG-16 network trained to classify images of scenes. (b) Dissecting emergent concepts in a Progressive GAN trained to synthesize images of kitchen scenes. Horizontal bars count the number of concepts that match units in a layer, out of a vocabulary of 1826 human-labeled visual concepts. In a classifier, units matching objects emerge in final layers of the network. In a generator, units matching objects emerge in the middle layers.

## From Instance Editing to Interactive Model Rewriting

The weights of a network encode relationships that we wish to understand. For instance, deactivating units for trees reveals the ability of a GAN to separately synthesize occluded buildings behind trees [3] (Figure 4a). Activating window units reveals the modeling of reflections (4b). Such interesting computations must be encoded in the weights; but how are the weights organized in the network? Can we directly manipulate them?

With regularization, weights in fine-grained layers of a network can be changed without affecting semantic computations. In SIGRAPH 2019 [6], I show how to apply this observation to enable semantic manipulation of a user-provided photograph including effects such as reflections.

In contrast, changing semantically interesting computations in middle layers requires a new perspective. My ECCV 2020 Model Rewriting work [4] develops a method to do this by viewing a deep layer in a trained network as an Optimal Linear Associative Memory (OLAM) [7, 8] that stores rules as a set of key-value pairs, memorized with minimal error.

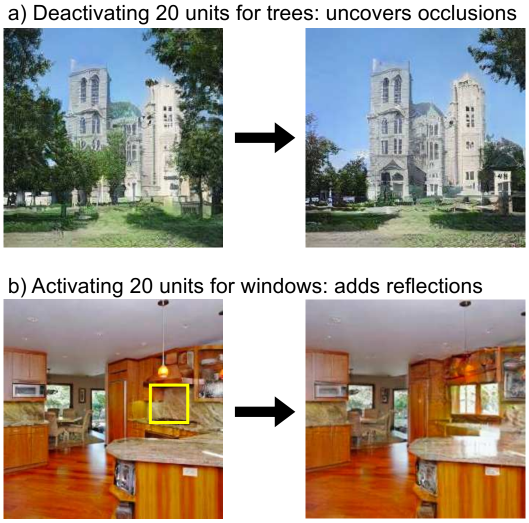
a) Deactivating 20 units for trees: uncovers occlusions
b) Activating 20 units for windows: adds reflections

Figure 4: (a) Erasing trees by deactivating 20 tree-causing units in `layer4` of ProgGAN [15] trained on LSUN churches [16] reveals that the GAN separately models buildings parts that were not visible. (b) Activating 20 window units in a kitchen model that we have modified to fit a specific photograph [6] shows GAN-synthesized reflections.

With this interpretation of a layer, changing a rule by rewriting a key-value pair leads to solvable equations where the optimal update is rank-one, with rows that are multiples of a vector $d$ that is independent of the value stored: this $d$ can be treated like a memory address. Even if a user chooses to change the stored value dramatically, sliding weights along $d$ will minimize interference with other rules.

This work allows users to directly express human agency in a model, enabling a user to make significant changes by rewriting the weights to create coherent images that are very different from the training distribution, without using any new training images. We have prototyped an interface to capture proposed edits. While conventional training does not generalize well, benchmarking different methods on recorded user interactions leads to our rank-one model rewriting method. Figure 5 shows how a user can edit a model based on their intentions using a copy-paste metaphor, pointing at a small number of examples to alter a rule that has a generalized impact on unseen outputs [4].

The work demonstrates how crossing the boundary between interactive interfaces and AI foundations can lead to new insights. By developing interfaces, methods, and empirical tests simultaneously, we can create new models of human-AI collaboration and find new machine learning approaches.

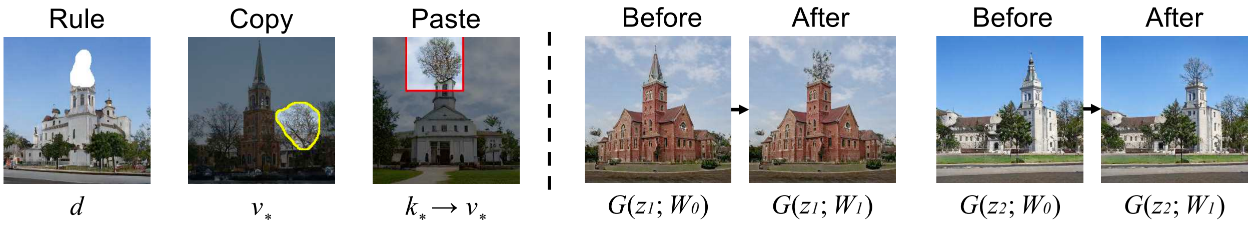

Figure 5: Interactively rewriting the rule for spires so they are generated as trees. Here we edit the weights of a StyleGANv2 [2] trained on LSUN churches [16]. (a) To rewrite the rule for spires, the user can scribble on any number of examples to pick the rule, then use copy-and-paste of a tree on a spire to demonstrate the desired change. (b) The altered model now draws trees instead of spires on unseen cases, while other aspects of the model remain unchanged.

## A Lab Devoted to Explainable AI

The problem of improving human control of opaque deep networks is an area of continuing concern across our industry. My work has been done in collaborations with IBM, DARPA, and Adobe, and has had some impact. (As the first demonstration of semantic paint using a GAN, GAN Paint was used by 60,000 people and forked 250 times, and it sparked follow-on semantic GAN painting methods from myself [6] as well as both Nvidia [17] and Google [18]; Network Dissection was cited 600 times, including a chapter in a textbook [19].)

As we apply AI across fields, it will be increasingly important for humans to understand and control the structures within deep networks. That will require collaboration and resources to develop interactive interfaces, to train and explain state-of-the-art models, and to develop explanatory data sets. Here are three initial challenges I would like my lab to take on.

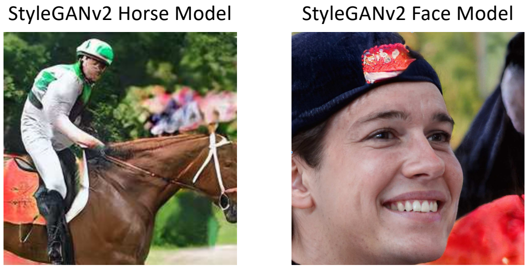

Figure 6: Left: a human face synthesized by a StyleGANv2 horse model is smeared and formless. Right: a StyleGANv2 trained on faces only is capable of drawing a nearly photorealistic face. Can we compose these learned models?

### 1. Composition

The face in the left image in Figure 6 is formless and far inferior to the model on the right. Could we improve the first model by composing the two networks? This is the essence of the problem of modular composition.

Connecting independently trained models is an unsolved problem that requires an interface that fills in the gaps for what the other model is incapable of representing, while preserving information the models have in common. We could train a composition model to join the two networks to improve quality of the output; or we could enable a human user to guide the composition. A starting point is the fundamental puzzle of how to compose a network with another version of itself. The goal is to enable collaboration and innovation through the creation of libraries of interchangeable network components.

### 2. Learning of Emergent Modules

How does a deep network learn that it should detect people wearing hats, or represent a horse leg with one neuron? The presence of concept detectors in deep networks has been observed in many contexts [20, 9, 10, 21, 22], but how such concepts emerge is not well-understood. Yet understanding emergent structure should be possible: deep network training is a fully transparent process, with every step, every gradient completely reproducible. The goal of this project is to map the mechanisms that cause emergence of modules. Based on this knowledge, we will develop methods to enhance this ability.

### 3. Deep Network Control Across Application Domains

Deep learning needs general tools for solving problems that go beyond imitation of the training data. The goal of this project is to generalize the methods we developed for Net Dissection, GAN Paint and Model Rewriting to non-image domains, such as NLP, robotics, and the use of deep networks in science. I look forward to collaborating with other university scientists to develop methods to identify and exploit machine-learned structure in new domains. We should be able to steer the topic of language models, and rewrite rules in RL policies. Can we help experts discover new quantities in a protein model?

The goal of my lab is to build foundational methods of a new type of software engineering, where human programmers and deep networks create algorithms collaboratively. To do this, we will develop methods that enable people to understand, reconfigure, and enhance the structure of learned computations.

## References

[1] Sylvia Fein and Heidi Scheuber. *Heidi's Horse*. Pleasant Hill, CA: Exelrod Press, 1976.

[2] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8110–8119.

[3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Zhou Bolei, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks". In: *ICLR*. 2019.

[4] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. "Rewriting a Deep Generative Model". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020.

[5] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. "Horses With Blue Jeans - Creating New Worlds by Rewriting a GAN". In: *Workshop on Machine Learning for Creativity and Design at NeurIPS*. 2020.

[6] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. "Semantic Photo Manipulation with a Generative Image Prior". In: *ACM TOG* 38.4 (2019).

[7] Teuvo Kohonen. "Correlation matrix memories". In: *IEEE transactions on computers* 100.4 (1972), pp. 353–359.

[8] James A Anderson. "A simple neural network generating an interactive memory". In: *Mathematical biosciences* 14.3-4 (1972), pp. 197–220.

[9] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Network dissection: Quantifying interpretability of deep visual representations". In: *CVPR*. 2017.

[10] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. "Understanding the role of individual units in a deep neural network". In: *Proceedings of the National Academy of Sciences* (2020).

[11] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." In: *ICCV*. 2017.

[12] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *ICLR*. 2015.

[13] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. "Learning deep features for scene recognition using places database". In: *NeurIPS*. 2014.

[14] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. "Seeing What a GAN Cannot Generate". In: *ICCV*. 2019.

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. "Progressive growing of gans for improved quality, stability, and variation". In: *ICLR*. 2018.

[16] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop". In: *arXiv preprint arXiv:1506.03365* (2015).

[17] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. "Semantic image synthesis with spatially-adaptive normalization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2337–2346.

[18] Andeep Singh Toor and Fred Bertsch. *Using GANs to Create Fantastical Creatures*. Nov. 2020. URL: `https://ai.googleblog.com/2020/11/using-gans-to-create-fantastical.html`.

[19] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2019.

[20] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[21] Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. "Identifying and Controlling Important Neurons in Neural Machine Translation". In: *NeurIPS*. 2018.

[22] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. "Causal mediation analysis for interpreting neural nlp: The case of gender bias". In: *arXiv preprint arXiv:2004.12265* (2020).