Transparent AI

Research Statement, David Bau

My research lies at the intersection of Computer Vision and Human-Computer Interaction.

As a long-time professional software engineer, 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 human programmers 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.

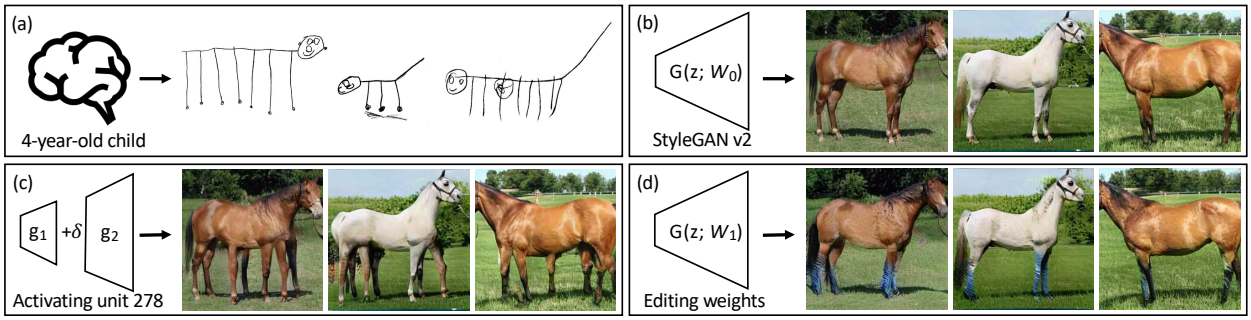

Figure 1: Comparing Heidi's horses [1] to a deep network. (a) Four-year-old Heidi's first horses reveal a modular understanding of legs and other parts. (b) Realistic horses generated by a StyleGANv2 [2] lead us to ask how the GAN represents legs. (c) Activating one unit inside the network adds legs 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 GAN Paint to manipulate the network's neurons like a paintbrush 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: Our 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. Rewriting treats network layers as memory matrices (aka OLAM [7, 8]), and lets a user change an individual memorized rule in the network, reprogramming the appearance of a leg, 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 people to directly control and reprogram deep networks by exploiting *emergent modularity within learned algorithms*.

Dissecting Deep Neworks

Why not impose modularity on a neural network by design? For example, if we need a system to classify photos of baseball fields and other sports, we could design the system with a specific module that recognizes baseballs or bats. Neural Module Networks [14] proposes this scheme, dividing a neural network into small modules with preset roles determined by the words of a question.

However, requiring programmers to prescribe the role for each module implies that the problem is solved on a conceptual level before the data is modeled. In contrast, allowing the network to learn its own organization exploits the network's ability to find new structure in the data. With the right tools, the network can teach us new ideas.

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 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 more ubiquitous in baseball while rare in other sports.

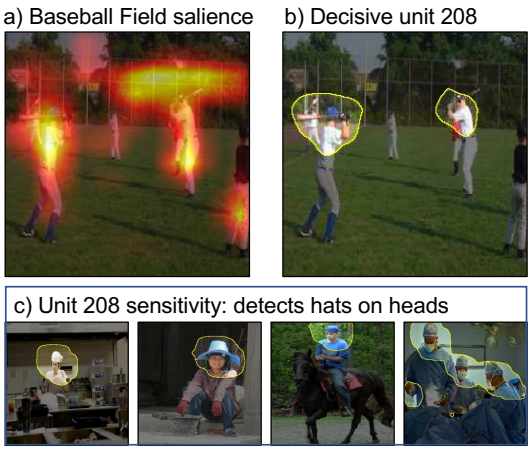

Figure 2: (a) The attention of a network can be decomposed into units (Grad-CAM [11] on a VGG scene classifier [12, 13] , classifying a baseball field). (b) The prediction for this image is insensitive to removal of most neurons in a layer `conv5_3`, but three units are decisive. The activation of one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

When trained to achieve high performance, neural networks can learn a broad array of useful concepts for solving a problem. Figure 3 (Bau et al, PNAS 2020 [10]) maps the vocabularies of emergent concepts that are learned by the layers of a classifier and a GAN generator. By developing methods to understand, harness, and build upon such emergent concepts, we can create a new type of software engineering that treats a neural network as a creative new member of the programming team.

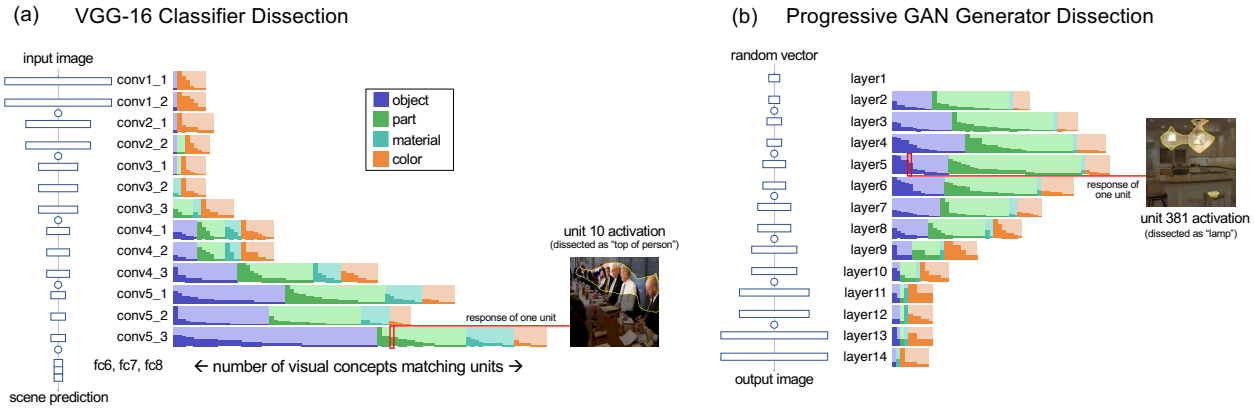

Figure 3: Using network dissection to compare emergent visual concepts in a classifier and a generator (PNAS 2020 [10]). (a) Dissecting emergent concepts in the layers of a VGG-16 network trained to classify images of scenes. (b) Dissecting emergent concepts in a Progressive GAN trained to synthesize 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.

A Lab Devoted to Transparent AI

The problem of improving human control of opaque deep networks is an area of continuing concern across our industry; my work to tackle the problem using emergent modularity has been done in collaborations with IBM, DARPA, and Adobe. Network dissection has over 600 citations including a chapter in a textbook [15]. The GAN Paint demo, the first demonstration of semantic paint using a GAN, was used by 60,000 people; its code was forked 250 times, and it sparked follow-on GAN painting methods from myself [6] (SIGGRAPH 2019) as well as both Nvidia [16] and Google [17].

Despite recent progress, the field of transparent AI is just getting started. As our community applies AI across fields, it will be increasingly important to understand and control modular structure within deep networks. Here are three initial challenges I would like my lab to take on.

Figure 4: At left, a human face synthesized by a StyleGANv2 horse model is smeared and formless; at 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 4 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.

To connect two models, we need an interface that fills in the gaps for what the other model is incapable of representing, while preserving the information that both models have in common. We can analyze and match representation statistics explicitly, or we could train a composition network on the task of joining the two networks to improve quality of the output. A starting point is the fundamental but unsolved problem of how to compose a network with an independently trained version of itself. Composition will be a powerful building block for engineering neural systems.

2. Control beyond Pixel Generators

Deep learning needs tools for solving problems where the behavior that we need goes beyond imitation of the training data. The goal of this project is to explore how the tools we developed for Net Dissection, GAN Paint and Model Rewriting can be applied to non-image domains, such as NLP, control, or the application of deep networks to science. I look forward to collaborating with other university scientists to develop methods to identify and exploit machine-learned structure in new domains. Can we steer the topic of a language model? Can we help experts discover new quantities within a protein model?

3. Modular Learning

How does a deep network learn that detection of people wearing hats is an important concept? The emergence of concept detectors in deep networks has been observed in many contexts [9, 10, 18, 19], but how such concepts appear is not well-understood. Yet understanding emergent structure should be possible: deep network training is a fully transparent process, with every step and every gradient completely reproducible. The goal of this project is to map the mechanisms that cause the emergence of modules. Based on this knowledge, we will develop methods to enhance this ability.

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, recombine, manipulate, 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] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. "Neural module networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 39–48.

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

[16] 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.

[17] 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`.

[18] 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.

[19] 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).