ConceptEvo: Interpreting Concept Evolution in Deep Learning Training

Haekyu Park [1] **Seongmin Lee** [1] **Benjamin Hoover** [1] **Austin Wright** [1] **Omar Shaikh** [2] **Rahul Duggal** [2]
Nilaksh Das [2] **Kevin Li** [1] **Judy Hoffman** [1] **Duen Horng (Polo) Chau** [1]

Abstract

We present CONCEPTEVO, a unified interpretation framework for deep neural networks (DNNs) that reveals the inception and evolution of learned concepts during training. Our work fills a critical gap in DNN interpretation research, as existing methods focus on post-hoc interpretation after training. CONCEPTEVO presents two novel technical contributions: (1) an algorithm that generates a unified semantic space that enables side-by-side comparison of different models during training; and (2) an algorithm that discovers and quantifies important concept evolutions for class predictions. Through a large-scale human evaluation with 260 participants and quantitative experiments, we show that CONCEPTEVO discovers evolutions across different models that are meaningful to humans and important for predictions. CONCEPTEVO works for both modern (ConvNeXt) and classic DNNs (e.g., VGGs, InceptionV3).

Figure 1. CONCEPTEVO creates a unified semantic space that represents and aligns concepts detected by neurons across different models in training (top: VGG19; middle: InceptionV3; bottom: ConvNeXt), embedding similar concepts (e.g., "furry dog," "car wheel," "mesh pattern") at similar corresponding locations.

1. Introduction

Interpreting how Deep Neural Networks (DNNs) arrive at their decisions has become crucial for instilling trust in models (Ribeiro et al., 2016), debugging them (Koh & Liang, 2017), and guarding against harms such as embedded bias or adversarial attacks (Das et al., 2020; Papernot & McDaniel, 2018; Zhang et al., 2018). As a fundamental type of DNN, convolutional neural networks (CNNs) have been intensively explored to understand their inner workings. For example, saliency-based methods identify important image regions for predictions (Selvaraju et al., 2017; Simonyan et al., 2013). Concept-based methods discover ***concepts*** detected by DNNs (e.g., "*furry dog*" concept in Fig 1) and their role in forming higher-level concepts and predictions (Park et al., 2021; Olah et al., 2020; Ghorbani et al., 2019; Kim et al., 2018; Bau et al., 2017).

However, existing interpretation approaches primarily focus on post-training analysis (Laugel et al., 2019; Guidotti et al., 2018), providing limited insights into the evolution of models during training. Crucially, understanding the progression of concepts detected by individual neurons, which we refer to as the neuron's **concept evolution**, and its association with model deficiencies like poor generalizability (Li et al., 2018; Zhang et al., 2021; Keskar et al., 2017) or convergence failures (Reddi et al., 2019; Arora et al., 2019) remains lacking. Relying solely on post-training interpretation poses challenges for real-time discovery and diagnosis during training, resulting in wasted time and resources (Elsken et al., 2019; Safarik et al., 2018), if the training ultimately fails to achieve the desired outcomes. Interpreting the model training process promotes effective monitoring (Zhong et al., 2017; Liu et al., 2017; Abadi et al., 2016; Zhou et al., 2022).

[1]Georgia Institute of Technology. [2]Work done while at Georgia Institute of Technology. Correspondence to: Haekyu Park <haekyu@gatech.edu>.

AI & HCI Workshop at the 40th International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA.

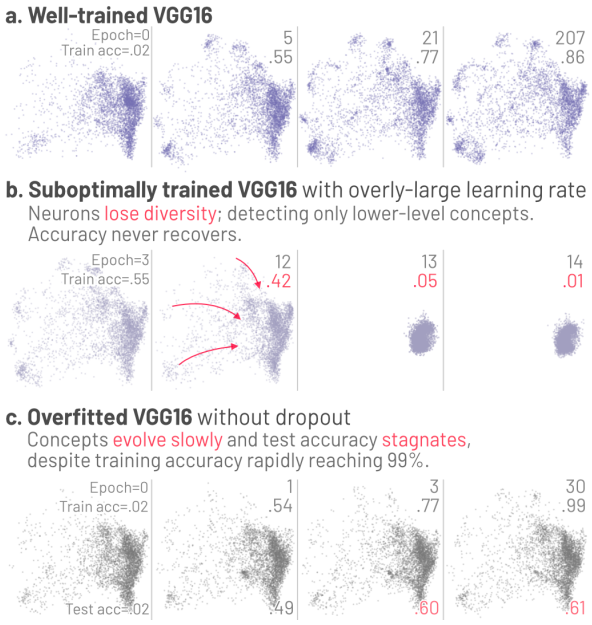

Figure 2. ConceptEvo identifies potential training issues. (a) A well-trained VGG16 shows gradual concept formations and refinements. (b) A VGG16 suboptimally trained with a large learning rate, rapidly losing the ability to detect most concepts. (c) An overfitted VGG16 without dropout layers, showing slow concept evolutions despite rapid training accuracy increases. We abbreviate "top-5 training/test accuracies" as "train/test acc."

To address the above research gaps, this work makes the following key contributions:

1. ConceptEvo: a unified DNN interpretation framework that reveals the inception and evolution of learned concepts during training (Sec 3). Our framework presents two novel technical contributions:

(1) Generating a unified semantic space that enables side-by-side comparison of different model training (Fig 1, 2). ConceptEvo is applicable for both modern (ConvNeXt) and classic DNNs (VGGs, InceptionV3).

(2) Discovering and quantifing important concept evolutions for a class prediction (Fig 3).

2. Extensive evaluation. We conduct large-scale human experiments with 260 participants and quantitative experiments to demonstrate that ConceptEvo discovers concept evolutions that are both meaningful to humans and important to class predictions (Sec 4).

3. Discoveries on model evolution. We highlight how ConceptEvo aids in uncovering potential issues during training and provides insights into their causes, such as: (1) incompatible hyperparameters (e.g., overly high learning rate) that severely harm concept diversity as shown in Fig 2b; and (2) slowly evolving concepts despite rapid increases in training accuracy in overfitted model as shown in Fig 2c.

2. Related Work

Interpreting DNNs After Training. Interpreting fully-trained DNNs revolves around describing features important to the model's behavior. For example, saliency-based methods identify image pixels that are important for predictions (Simonyan et al., 2013; Simonyan & Zisserman, 2015; Gan et al., 2015; Selvaraju et al., 2017). However, saliency-based interpretation faces a challenge as important image pixels may not align with high-level concepts that are easily understandable to humans (Kim et al., 2018; Gulshad & Smeulders, 2020). To address this, recent studies have focused on explaining high-level, human-understandable concepts learned within the model and their relevance to the model's prediction (Hernandez et al., 2022; Ghorbani & Zou, 2020; Yeh et al., 2020; Goyal et al., 2019; Zhou et al., 2018; Kim et al., 2018; Nguyen et al., 2016). For example, feature visualization techniques (Zeiler & Fergus, 2014; Yosinski et al., 2015) generate synthetic images that strongly activate specific neurons, visualizing detected concepts. ACE (Ghorbani et al., 2019) discovers important super-pixel image segmentations, presenting learned concepts important for predictions. Net2Vec (Fong & Vedaldi, 2018) encodes individual neurons' concepts into vectors by using predefined concept images. NeuroCartography (Park et al., 2021) visualizes concepts detected by neurons through encoding conceptual neighborhood of neurons.

Interpreting DNNs During Training. Many existing studies to interpret DNNs *during training* focus on how data representations within DNNs evolve across epochs and how the evolution impacts its downstream performance (Pühringer et al., 2020; Smilkov et al., 2017; Chung et al., 2016). DeepEyes (Pezzotti et al., 2017) studies the evolution of individual neurons' activation for different classes during training. DGMTracker (Liu et al., 2017) analyzes the changes of weights, activations, and gradients over time. Others track the 2D projected evolution of neurons towards or away from certain labels (Rauber et al., 2016; Li et al., 2020), which limits our understanding of a learned concept to only available labels. DeepView (Zhong et al., 2017) introduces metrics to estimate whether neurons are evolving sufficient diversity for classification. ConceptEvo distinguishes itself from the existing approaches by enabling comparison of concepts learned by neurons from **any layers within a model** and even neurons from a **different model** trained using the same dataset.

3. Method

3.1. Desiderata of Interpreting Concept Evolution

D1 General interpretation of concept evolution across different models. Comparing different model training is indispensable for deciding which model is trained bet-

ter or which training strategy is more effective (Li et al., 2018; Raghu et al., 2017). Thus, we aim to develop a general method that enables side-by-side comparison and interpretation of concept evolution across different models. (Sec 3.2)

D2 Revealing and quantifying important evolution of concepts. We aim to identify internal changes that significantly impact the prediction of a specific class, as understanding the most influential components can lead to effective model improvements (Ghorbani & Zou, 2020). For example, how important is the evolution of a neuron's concept (e.g., from "brown color" to "brown furry leg") for the prediction of a class (e.g., "brown bear")? We aim to automatically find such important changes in concepts for a class prediction. (Sec 3.3)

D3 Discoveries. Can interpreting model evolution help identify training problems and offer insights for addressing them, advancing prior work that focuses on interpreting and fixing models post-training (Ghorbani & Zou, 2020)? For example, can we help determine if a model's training is on track and if interventions are required to improve accuracy? (Sec 4.5)

3.2. General Interpretation of Concept Evolution

To compare the evolution of models, we face the challenge of aligning concepts across different models and training stages. Different models are independently trained; the learned concepts are not aligned by default. Even for the same model, activation patterns can change considerably over training epochs.

To address this challenge, we propose a two-step method. In step 1, we create a base semantic space that captures concepts identified by a *base model* at a specific training epoch. This semantic space serves as a reference for concept representation. In step 2, we project concepts from other models at all epochs onto the base semantic space, creating a **unified semantic space** where similar concepts across different models and epochs are mapped to similar locations.

To ensure comprehensive concept coverage, we select an optimally trained model as our base model. For example, we used a fully trained VGG19 (Simonyan & Zisserman, 2015) as a base model for Fig 1 and 2.

Step 1: Creating the base semantic space. We use neurons as units to represent concepts, leveraging their selective activation for specific concepts (Ghorbani & Zou, 2020; Olah et al., 2017; Yosinski et al., 2015). By using neurons, we can pinpoint areas of interest in models, enabling focused troubleshooting, particularly in identifying abnormal training patterns within specific groups of neurons. Building on prior work (Park et al., 2021), we embed neurons that strongly respond to common inputs in similar locations. As neuron-concept relationships may not always be one-to-one (Olah et al., 2020; Fong & Vedaldi, 2018), we aim to generalize to many-to-many relationships. For example, polysemantic neurons responsive to multiple concepts are embedded between those concepts.

Step 1.1: Find stimuli. For each neuron, we collect k images that evoke the highest activation in the neuron's activation map, which we refer to as the neuron's stimuli. Neurons associated with a single concept have similar stimuli, while polysemantic neurons may have stimuli representing multiple concepts.

Step 1.2: Sample frequently co-activated neuron pairs. We create a multiset D of neuron pairs that are strongly co-activated in the base model M_b at epoch t_b. For each image $\mathbf{x}$, we create a list of neurons that are strongly co-activated by $\mathbf{x}$, by collecting neurons with $\mathbf{x}$ in their stimuli. We randomly shuffle each list of co-activated neurons and sample neuron pairs using a sliding window of length two. The sampled neuron pairs are then added to D. This sampling process is repeated E times to obtain diverse neuron pairs. A specific neuron pair can appear multiple times in D, with higher frequency as more images are shared by their stimuli. This leads to a closer embedding of frequently co-activated neurons in the unified semantic space.

Step 1.3: Learn neuron embedding. The objective function, defined by Eq (1), represents a negative log likelihood to learn neuron embeddings; intuitively, (1) co-activated neuron pairs with a larger inner product (and spatially closer embeddings) are more likely to indicate similar concepts, while (2) randomly paired neurons with a lower inner product) and spatially farther embeddings) are less likely to be conceptually similar. The randomly paired neurons serve as negative examples, enabling high-quality vector representations of concepts, similar to the negative sampling approach used in Word2Vec algorithm (Mikolov et al., 2013a;b). This neuron embedding approach allows for the representation of many-to-many relationships between neurons and concepts. For example, a polysemantic neuron, which is co-activated by multiple distinct groups of neurons representing different concepts, is attracted towards these groups, resulting in its spatial location between them. In the objective function, $\mathbf{v}^t_{n,M}$ is an embedding of neuron n in model M at epoch t. r is a randomly selected neuron. R is the number of randomly sampled neurons for each co-activated neuron pair in D. $\sigma(\cdot)$ is the sigmoid function (i.e., $\sigma(x) = 1/(1+e^{-x})$).

$$J_1 = -\sum_{(n,m)\in D}\Bigg(\log\left(\sigma(\mathbf{v}^{t_b}_{n,M_b}\cdot\mathbf{v}^{t_b}_{m,M_b})\right) + \sum_{r=1}^{R}\log\left(1-\sigma(\mathbf{v}^{t_b}_{n,M_b}\cdot\mathbf{v}^{t_b}_{r,M_b})\right) + \sum_{r=1}^{R}\log\left(1-\sigma(\mathbf{v}^{t_b}_{m,M_b}\cdot\mathbf{v}^{t_b}_{r,M_b})\right)\Bigg) \tag{1}$$

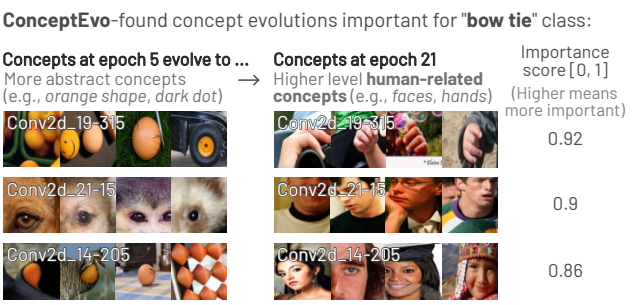

Figure 3. CONCEPTEVO identifies and quantifies important concept evolutions for class prediction. For example, in a VGG16 model, it discovers important evolutions such as abstract to human-related concepts that contribute to predicting the "bow tie" class. The evolution in the top row has a score of 0.92, meaning that 92% of bow tie images benefit from this evolution.

Step 2: Unifying the semantic space of different models and epochs. If two neurons, regardless of their corresponding models or training stages, have similar stimuli (i.e., they are strongly co-activated by many common input images), we aim to learn their embeddings to be close together in the unified semantic space. However, learning all concepts in all models and epochs simultaneously in Step 1 is computationally impractical. Thus, we introduce Step 2 to approximately represent concepts not present in the base model.

Step 2.1: Image embedding. Different models, despite their varying architectures and neurons, share a common characteristic of being trained on the same dataset. Using this commonality, we consider two neurons from different models to detect similar concept if they are both activated by similar inputs. To learn neuron embeddings across models, we use image embeddings as a bridge. As an image includes various concepts, we assume that an image vector can be formed by linearly combining the vectors of those concepts. Conversely, we assume that a neuron's embedding can be approximated by linearly combining image embeddings. We approximate a neuron's embedding by averaging the embeddings of the images in its stimuli, as average is a standard aggregation approach in prior research (Ghorbani & Zou, 2020; Ghorbani et al., 2019; Kim et al., 2018). Specifically, we learn image embedding by minimizing the L2 loss between the approximated neuron embedding and the neuron embedding in the base model.

Step 2.2: Approximate embedding of neurons in a non-base model. After embedding images in Step 2.1, we approximate neuron embeddings of other models at other epochs by averaging embedding of images in the neuron's stimuli. Step 2.2 is the only (sub)step that needs to be performed when projecting a new model onto the unified semantic space. There is no need to re-run Step 1 and 2.1.

To visualize neuron embeddings, we use UMAP, a non-linear dimensionality reduction method that preserves both the global data structures and local neighbor relations (McInnes et al., 2018). To assist in understanding the concepts detected by each neuron, we compute example patches which are cropped images that highly activate the neuron (e.g., example patches of neurons for *"furry dog"* concept in Fig 1) (Olah et al., 2017).

3.3. Concept Evolutions Important for a Class

As discussed in **D2**, our objective is to uncover concept evolutions that are crucial for class predictions. For example, how important is the evolution of a neuron's concept (e.g., from *"furry animals' eyes"* to *"human neck"*) to the prediction for a class (e.g., "bow tie")? Inspired by (Kim et al., 2018), we quantify the importance of a concept evolution by evaluating how sensitive a class prediction is to the evolutionary state of the concepts.

Let n be a neuron in layer l of model M, and $Z^t_{n,l,M}(\mathbf{x})$ be its activation map at epoch t given image $\mathbf{x}$. The function $h^t_{l,c}(\cdot) : \mathbb{R}^{h_l \times w_l \times s_l} \rightarrow \mathbb{R}$ takes activation of l as input and returns the logit value for class c, where h_l, w_l, and s_l are height, width, and the number of neurons in l, respectively. Let $Z^t_{l,M}(\mathbf{x}) \in \mathbb{R}^{h_l \times w_l \times s_l}$ be the activation of layer l in model M given the input $\mathbf{x}$, and $\Delta Z^{t,t'}_{n,l,M}(\mathbf{x})$ is the activation change of n from epoch t to t' as defined in Eq (2). $\mathbf{0}_{a,b}$ is a zero matrix of a rows and b columns. Eq (3) defines the sensitivity of the class c prediction with respect to n's concept evolution from t to t' given $\mathbf{x}$. The directional derivative in Eq (3) indicates how sensitively a prediction for class c would change if the activation in layer l changes towards the direction of neuron n's evolution. A positive value indicates that the concept evolution of neuron n positively contributes to the prediction for class c.

$$\Delta Z^{t,t'}_{n,l,M}(\mathbf{x}) = [\mathbf{0}_{h_l,w_l}, ..., \underbrace{Z^{t'}_{n,l,M}(\mathbf{x}) - Z^{t}_{n,l,M}(\mathbf{x})}_{n\text{-th matrix}}, ..., \mathbf{0}_{h_l,w_l}] \tag{2}$$

$$\begin{aligned} S^{t,t'}_{n,l,M,c}(\mathbf{x}) &= \lim_{\epsilon \to 0} \frac{h^t_{l,M,c}\big(Z^t_{l,M}(\mathbf{x}) + \epsilon \Delta Z^{t,t'}_{n,l,M}(\mathbf{x})\big) - h^t_{l,M,c}\big(Z^t_{l,M}(\mathbf{x})\big)}{\epsilon} \\ &= \nabla h^t_{l,M,c}(Z_l(\mathbf{x})) \cdot \Delta Z^{t,t'}_{n,l,M}(\mathbf{x}) \end{aligned} \tag{3}$$

We finally measure the importance of concept evolution of a neuron n in layer l in model M from epoch t to t' for class c, by aggregating the importance across class c images, as in Eq (4), where X_c is the set of images labeled as c.

$$I^{t,t'}_{n,l,M,c} = \frac{|\{\mathbf{x} \in X_c \ : \ S^{t,t'}_{n,l,M,c}(\mathbf{x}) > 0\}|}{|X_c|} \tag{4}$$

Fig 3 illustrates important concept evolutions for the "bow tie" class discovered by CONCEPTEVO, such as evolutions from abstract concepts to *"human hands," "neck,"* and *"face"* concepts. Surprised by the many evolutions to human-related concepts, we inspected the raw images for the bow tie class and confirmed that most of the images (over 70%) are of a person wearing a bow tie.

4. Experiment

We evaluate how well CONCEPTEVO satisfies the desired properties for interpreting conception evolution (Sec 3.1, D1-3) by answering the following research questions:

- **Q1 Alignment.** How well does CONCEPTEVO's neuron embedding align concepts of different models at different training stages in the unified semantic space? (Sec 4.2, for **D1**)
- **Q2 Meaningfulness.** How are the discovered concept evolutions semantically meaningful? (Sec 4.3, for **D1**)
- **Q3 Importance.** How are the discovered concept evolutions important for class prediction? (Sec 4.4, for **D2**)
- **Q4 Discoveries.** How does CONCEPTEVO provide insightful discoveries? (Sec 4.5, for **D3**)

4.1. Experiment Settings

Datasets and models. We study concept evolutions in representative image classifiers trained on ILSVRC2012 (ImageNet) (Russakovsky et al., 2015), including a modern model like ConvNeXt (Liu et al., 2022) inspired by recent models such as ResNet (Targ et al., 2016), ResNeXt (Xie et al., 2017), and vision transformers (Liu et al., 2021; Kolesnikov et al., 2021), as well as classic models (e.g., VGG16 (Simonyan & Zisserman, 2015), VGG19 (Simonyan & Zisserman, 2015), VGG16 without dropout layers (Srivastava et al., 2014), and InceptionV3 (Szegedy et al., 2016)). To ensure comparable accuracies, we trained the models using hyperparameters from prior work (Simonyan & Zisserman, 2015; Szegedy et al., 2016; Liu et al., 2022).

Hyperparameter settings. We carefully selected hyperparameters to achieve the overarching goal of a unified semantic space that balances strong coherence among neighboring neurons with computation efficiency. Specifically, the following hyperparameter values were tested within the indicated ranges: the number of stimuli per neuron (k) was tested from 5 to 30, with a chosen value of 10 to strike the balance; the dimension of neuron and image embeddings was set to 30 (tested from 5 to 100); the learning rate for neuron embedding was set to 0.05 and for image embedding, it was set to 0.002 (tested from 0.001 to 0.5); and the number of randomly sampled neurons per neuron pair (R) was set to 3 (tested from 0 to 5).

4.2. Alignment of Neuron Embeddings

To ensure the effectiveness of CONCEPTEVO in aligning concepts across models and epochs, we evaluate the conceptual coherency of neighboring neurons on the unified semantic space through a large-scale human evaluation with Amazon Mechanical Turk (MTurk), modeling after prior work (Park et al., 2021; Ghorbani et al., 2019). The evaluation focuses on four categories: (1) hand-picked sets of neurons of similar concepts, serving as a baseline; (2) neuron groups detected by CONCEPTEVO from the base model (a well-trained VGG16); (3) neuron groups in the same model at different training epochs, detected by CONCEPTEVO; (4) neuron groups from different models (VGG16 and InceptionV3) at different epochs, detected by CONCEPTEVO. We collect the neuron groups by running K-means clustering on the neuron embeddings on the unified semantic space.

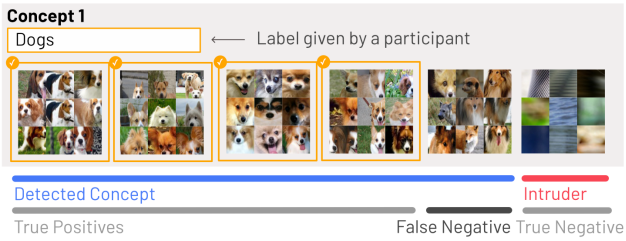

Figure 4. MTurk questionnaire example. Participants are shown six neurons' example patches and asked to determine if there is a semantically coherent group among them. If so, they provide a short label for the group. In the example, the first five neurons are semantically similar, detected and grouped by CONCEPTEVO; the rightmost is randomly sampled. Here, a participant correctly identifies the first four neurons as a coherent "dogs" concept (four *true positives*), misses the fifth neuron (one *false negative*), and correctly identifies the intruder as unrelated (one *true negative*).

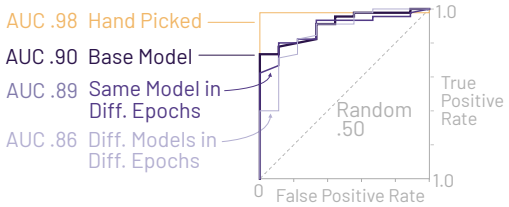

Figure 5. ROC Curve for human estimations demonstrating the high alignability of concepts discovered by CONCEPTEVO, even when sampled across different models and epochs.

We conducted concept classification tasks with 260 MTurk participants, presenting them with nine unique tasks. Each task includes six neurons, randomly ordered, where five of them had similar concepts identified by CONCEPTEVO or were hand-picked, while one was a randomly selected "intruder" neuron. Participants were provided with nine example image patches for each neuron to aid in understanding its concept. Participants were not informed about the number of potential intruders and were asked to select as many neurons as they believed to be semantically similar, and were asked to provide a brief description of the concept. This process, illustrated in Fig 4, essentially forms a classification task, treating the participants as classifiers and the grouped neurons as true labels. This generated a test set with a total of 10,950 individual determinations of conceptual inclusions for neurons. From this framing, we consider success by how consistently participants agree with the model determination. Fig 5 shows an ROC curve with the participants' determinations, demonstrating the high discernibility and alignment of CONCEPTEVO-detected concepts. Even when sampling concepts across different epochs and models, the AUC scores were consistently high, ranging from 0.90

for sampling within the base model to 0.86 for sampling across different models and training epochs.

4.3. Meaningfulness of Concept Evolution

Concepts discovered by CONCEPTEVO should be *meaningful* and informative to humans. We evaluate the *interpretive consistency* of the concepts labeled and described by the participants (Fig 4). To handle variations in phrasing for the labels, we use sentence-level embeddings from the Universal Sentence Encoder (USE) (Cer et al., 2018). USE captures the semantic similarity of phrases such as "vehicle wheels," "cars," and "trucks." To establish a baseline for similarity, we calculate the average pairwise similarity between all labels, resulting in 0.28. Subsequently, we measure the average pairwise similarity between the labels provided by participants for individual concepts within each category from 4.2. The results are as follows: (1) the average concept similarity for hand-picked concepts is 0.455, (2) the average concept similarity for concepts from the base model is 0.40, (3) the average concept similarity for concepts within the same model but different epoch is 0.40, and (4) the average concept similarity for concepts from different models and different epochs is 0.38. All of these values significantly exceed the baseline similarity of 0.28, indicating that concepts discovered through CONCEPTEVO are reliable and meaningful, even when assessed by different people.

4.4. Concept Evolutions Important to a Class

CONCEPTEVO quantifies and identifies important concept evolutions, as illustrated in Fig 6. In InceptionV3, it reveals evolutions from abstract to bird-related concepts for the "Goldfinch" class. Similarly, in ConvNeXt, it discovers evolutions from abstract to dog-related concepts for the "Shetland sheepdog" class. Some neurons become more specialized during training, as exemplified by a neuron evolving from detecting a *dark background* to a *dark-eyed circle* and eventually to a *head with a dark eye* (the first row of Fig 6).

To evaluate CONCEPTEVO's effectiveness in identifying important concept evolutions, we measure accuracy changes when reverting evolutions, following prior work that evaluate concept importance in fully-trained models (Ghorbani & Zou, 2020; Ghorbani et al., 2019). We revert the activation map of a neuron from a future epoch t' to a past epoch t and measure the accuracy at t'. By comparing the accuracy before and after reverting, we gain insights into the importance of the concept evolution: a larger drop in accuracy indicates a higher importance for the concept evolution of that neuron. To determine the stages of evolution to evaluate, we identify the epochs with top-1 training accuracies closest to the milestones of 25%, 50%, and 75%. For VGG16, the evolution stages are 5→21 and 21→207; for InceptionV3, 1→11 and 11→121; and for ConvNeXt, 1→3 and 3→96.

As CONCEPTEVO measures the importance of concept evolution for a single neuron (Eq 4), it is natural to evaluate

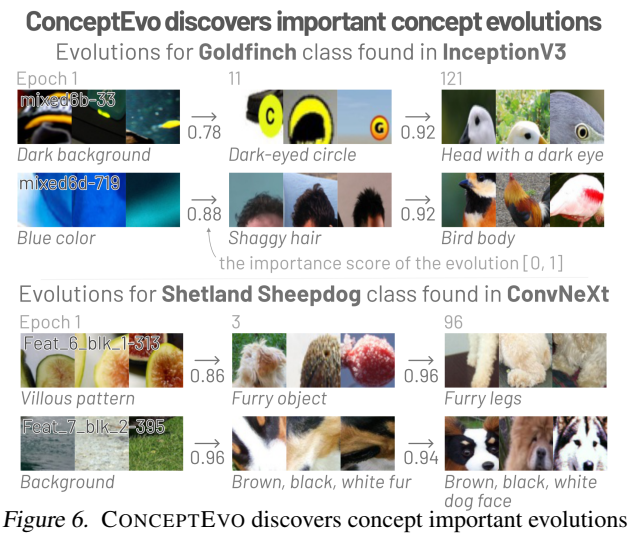

Figure 6. CONCEPTEVO discovers concept important evolutions for class predictions. For example, it identifies dog-related evolutions important for the "Shetland sheepdog" class in ConvNeXt, and bird-related evolutions important for the "Goldfinch" class in InceptionV3.

accuracy changes by reverting each neuron's evolution individually and then aggregating the changes. However, due to the large number of neurons, this approach becomes computationally prohibitive. To address this, we propose a more practical approach: reverting multiple evolutions in a layer simultaneously and aggregating accuracy changes across layers. The evaluation process involves five steps for each class c and evolution stage from epoch t to t'. **Step 1:** Sample 128 images (around 10% of images for class c). **Step 2:** Compute the importance of concept evolutions for all neurons using Eq 4. **Step 3:** Rank neurons in each layer by the importance and divide them into four importance bins: 0-25th percentile (most important), 25-50th percentile, 50-75th percentile, and 75-100th percentile. **Step 4:** Revert evolutions of neurons in each bin, compute accuracy at t', and measure the accuracy changes compared to the non-reverted accuracy. **Step 5:** Average accuracy changes across layers for each bin. To mitigate sampling bias in Step 1, we independently repeat the procedure five times. We then average the accuracy changes across 100 randomly selected classes from the 1,000 classes in ImageNet[1].

Fig 7 shows the impact of reverting evolutions in different importance bins on the top-1 training accuracy of VGG16, ConvNeXt, and InceptionV3. Reverting higher-importance evolutions (lower percentiles) leads to larger accuracy drops, confirming CONCEPTEVO's effectiveness in identifying important concept evolutions. Interestingly, reverting the least important evolutions (75-100th percentile) sometimes leads to increased accuracy. This suggests that least important

[1]Standard deviations of the average accuracy changes across the classes between the five runs are very low (e.g., 9.2e-5 for top-1 training accuracy and 2.1e-4 for top-1 test accuracy, for the 21→207 evolution).

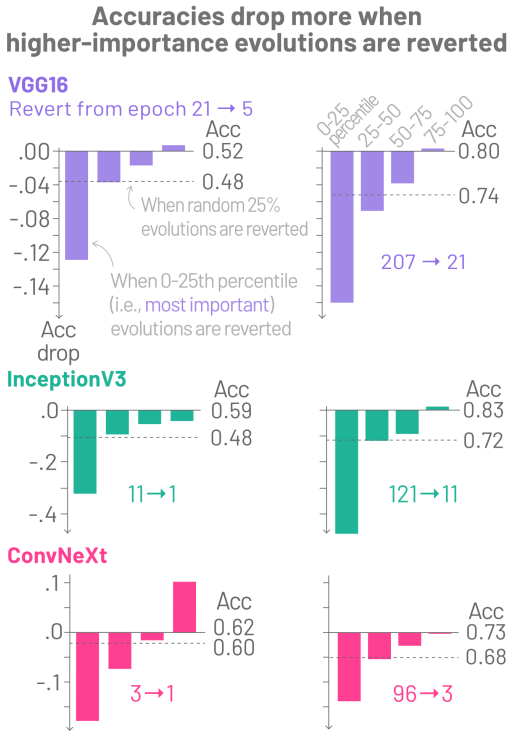

Figure 7. Evaluation of CONCEPTEVO's ability to identify important concept evolutions for 100 random classes. Neurons are ranked by their evolution importance and divided into four bins: 0-25th (most important), 25-50th, 50-75th, 75-100th percentiles. Reverting higher-importance evolutions leads to a larger accuracy drop, demonstrating CONCEPTEVO's effectiveness in identifying important concept evolutions. As a baseline, we compared the accuracy drop when randomly reverting 25% (i.e., the same number of neurons in each bin) evolutions, which falling between the 25-50th and 50-75th percentile bins.

evolutions may interfere with the corresponding class predictions. As a baseline, we reverted 25% randomly selected evolutions, resulting in an accuracy drop between the 25-50th percentile and 50-75th percentile. Furthermore, we evaluated the changes in the top-5 training, top-1 test, and top-5 test accuracies when reverting evolutions in the same four bins, reinforcing our key finding that reverting higher-importance evolutions results in larger accuracy drop.

4.5. Discovery

Incompatible hyperparameters harm concept diversity. CONCEPTEVO's neuron concept embedding helps identify issues caused by incompatible hyperparameters and provides insights into their impact on model performance. For example, in Fig 2b, CONCEPTEVO reveals that a VGG16 suboptimally trained with an excessively high learning rate[2] experiences a drastic accuracy drop during training. Also, atrophying of neuron concepts degrades concept diversity and captures only lower-level concepts, which is apparent even before the accuracy reaches 0. The loss of diversity

[2]0.05, larger than the optimal 0.01 presented in prior work

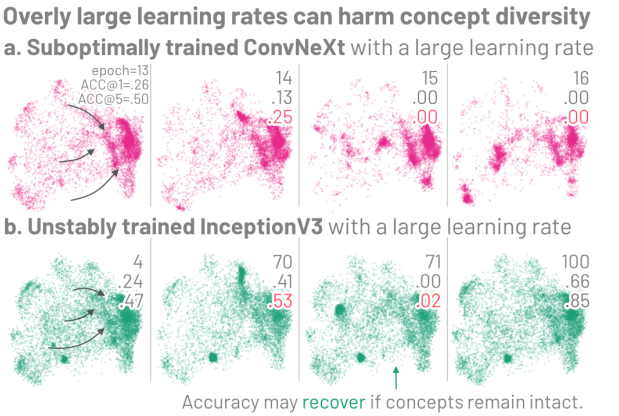

Figure 8. A suboptimally trained ConvNeXt and an unstably trained InceptionV3 with large learning rates experience decreased concept diversity and convergence in a few regions (e.g., right side) to detect lower-level concepts, specifically when these models' training accuracies drop (the second and third columns). Interestingly, the training accuracy of InceptionV3 recovers, as concepts remain at epoch 71.

is so severe that it cannot be recovered even with 40 more epochs. A similar pattern is observed in a ConvNeXt model trained with a high learning rate[3], as shown in Fig 8a.

In the case of an InceptionV3 unstably trained with a large learning rate[4], CONCEPTEVO reveals a similar yet slightly different scenario. As depicted in Fig 8b, the accuracy significantly drops at epoch 71, but interestingly, it recovers after a few more epochs. This recovery is likely due to the persistence of a large number of concepts at epoch 71, despite the low accuracy.

These examples demonstrate that CONCEPTEVO can provide actionable insights to determine whether interventions, such as stopping the training, might be desirable. Severe damage to concept diversity, as observed in Fig 2b and 8a, suggests that stopping the training might be more beneficial, as the model is unlikely to recover even with further epochs, compared to the milder damage depicted in Fig 8b.

To quantitatively study concept diversity in models, we employ differential entropy which measures the uncertainty in a continuous variable (Michalowicz et al., 2013). We compute the differential entropy for each dimension of neuron embeddings and average the values across the dimensions[5]. Higher values indicate more diverse concepts. In a VGG16 suboptimally trained with a large learning rate (Fig 2b), the differential entropy decreases: 2.03→1.89→-1.52→-1.52 for epochs 3, 12, 13, 14, indicating a loss of concept diversity. Similarly, in a suboptimally trained ConvNeXt (Fig 8a), the differential entropy decreases: 1.64→1.57→1.54→1.34

[3]0.02, larger than the optimal 0.004 used in prior work

[4]1.5, larger than the optimal 0.045 used in prior work

[5]We average differential entropy across reduced 2D embeddings instead of the original dimension, since computing the entropy for high dimensional vectors leads to infinity.

for epochs 13, 14, 15, 16. In contrast, optimally trained models show increasing differential entropy, indicating that concepts become more diverse over epochs. For example, in an optimally trained VGG16 (Fig 2a), the differential entropy increases: 1.60→1.97→2.06→2.09 for epochs 0, 5, 21, 207. In the case of an unstably trained InceptionV3 (Fig 8b), the differential entropy decreases until epoch 71 (lowest accuracy) and then rebounds: 1.89→1.85→1.73→1.91 for epochs 4, 70, 71, 100, indicating that its concept diversity was initially damaged but later restored.

Overfitting slows concept evolution. Overfitting is a common issue in DNN training (Rice et al., 2020; Cogswell et al., 2016). Using CONCEPTEVO, we have discovered that concepts in overfitted models evolve at a slower pace, despite experiencing rapid increases in training accuracy. To intentionally induce overfitting, we modified a VGG16 (Fig 2c) by removing its dropout layers which are known to help mitigate overfitting (Srivastava et al., 2014). Additionally, we overfit a ConvNeXt model by setting the weight decay of the AdamW optimizer to 0, reducing its regularization effect (Loshchilov & Hutter, 2017). These models are overfitted expectedly[6].

We observed that overfitted models exhibit slower concept evolution compared to well-trained models. To increase the top-1 training accuracy from approximately 0.25 to 0.5 and from approximately 0.5 to 0.75, the neuron embeddings in a well-trained VGG16 model (Fig 2a) move an average Euclidean distance of 1.34 and 1.30, respectively. In contrast, the overfitted VGG16 model (Fig 2b) exhibits much slower movement, with neuron embeddings only shifting by 1.11 and 0.98 for the same accuracy increments. Similarly, for the well-trained ConvNeXt model, raising the top-1 training accuracy from approximately 0.25 to 0.5 and from approximately 0.5 to 0.75 corresponds to neuron embeddings moving an average distance of 1.39 and 1.42, respectively. Conversely, the overfitted ConvNeXt model shows slower movement, with neuron embeddings shifting by only 1.35 and 1.39 for the same accuracy increments.

4.6. Comparison with Existing Approaches

We compare CONCEPTEVO with existing methods in representing evolving concepts. Existing methods are not optimized to capture changes across epochs; they can be applied to one epoch at a time, independent of other epochs. We compare CONCEPTEVO with NeuroCartography (Park et al., 2021) and ACE (Ghorbani et al., 2019). ACE represents concepts with image segments that activate a layer. We use the final layer as described in the work and image segments from Broden dataset (Bau et al., 2017). We use UMAP (McInnes et al., 2018) for 2D visualization of concepts, running it for all epochs simultaneously to avoid misalignment caused by independent epoch-based reduction.

[6]In VGG16, at epoch 30, its top-1 train, top-5 train, top-1 test, top-5 test accuracies are 0.99, 1, 0.37, 0.61, respectively. In ConvNeXt, at epoch 32, its top-1 train, top-5 train, top-1 test, top-5 test accuracies are 0.94, 0.99, 0.57, 0.80, respectively.

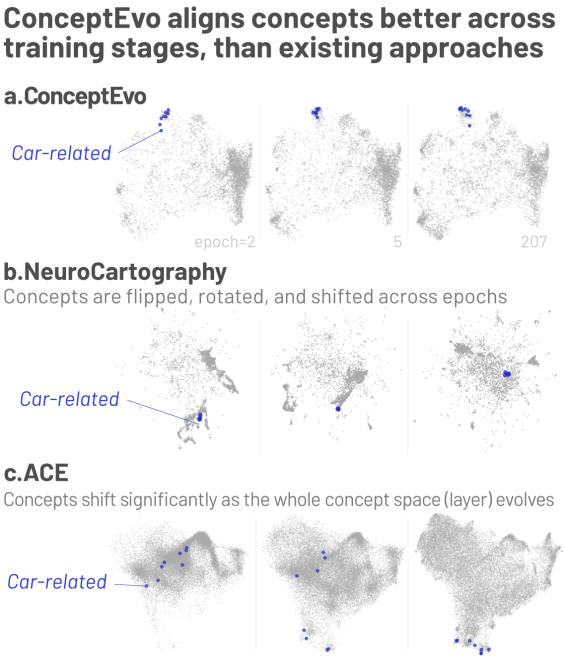

Figure 9. We compare the representation of concepts in VGG16 using ConceptEvo with existing methods. (a) CONCEPTEVO effectively aligns learned concepts across training epochs, by projecting similar concepts to similar locations. (b) Concepts represented by NeuroCartography exhibit flipping, rotation, and shifting across epochs, indicating misalignment. (c) Concepts represented by ACE undergo shifting, as the entire concept space (layer activation space) changes during training, indicating misalignment as well.

CONCEPTEVO aligns concepts well across epochs, while existing methods show misalignment. In Fig 9a, the "car-related" concept neurons are consistently located at the top left in epochs 2, 5, and 207. In Fig 9b, the "car-related" neurons represented by NeuroCartography are flipped, rotated, and shifted across epochs. In Fig 9c, the "car-related" image segments represented by ACE exhibit significant shifting as the concept space changes during training.

5. Conclusion and Future Work

CONCEPTEVO is a unified interpretation framework for DNNs that reveals concept inception and evolution during training. Through both large-scale human experiments and quantitative experiments, we have demonstrated that CONCEPTEVO can discover concept evolutions that aid human interpretation of model training across various models. This assists in identifying training issues and guiding interventions for more stable and effective training. In future work, we plan to extend our investigation to other types of models (e.g., object detectors and language models).

Acknowledgements

We thank our colleagues, reviewers, and the support from DARPA GARD, JPMorgan fellowship, and Cisco.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. {TensorFlow}: A system for {Large-Scale} machine learning, 2016.

Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks. *International Conference on Learning Representations (ICLR)*, 2019.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pp. 169–174, 2018.

Chung, S., Park, C., Suh, S., Kang, K., Choo, J., and Kwon, B. C. Revacnn: Steering convolutional neural network via real-time visual analytics. In *Future of interactive learning machines workshop at the 30th annual conference on neural information processing systems (NIPS)*, 2016.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., and Batra, D. Reducing overfitting in deep networks by decorrelating representations. *The International Conference on Learning Representations (ICLR)*, 2016.

Das, N., Park, H., Wang, Z. J., Hohman, F., Firstman, R., Rogers, E., and Chau, D. H. P. Bluff: Interactively deciphering adversarial attacks on deep neural networks. *IEEE Visualization Conference*, 2020.

Elsken, T., Metzen, J. H., and Hutter, F. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

Fong, R. and Vedaldi, A. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 8730–8738. Computer Vision Foundation / IEEE Computer Society, 2018.

Gan, C., Wang, N., Yang, Y., Yeung, D.-Y., and Hauptmann, A. G. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2568–2577, 2015.

Ghorbani, A. and Zou, J. Y. Neuron shapley: Discovering the responsible neurons. *Advances in Neural Information Processing Systems*, 33:5922–5932, 2020.

Ghorbani, A., Wexler, J., Zou, J., and Kim, B. Towards automatic concept-based explanations. *Neural Information Processing Systems*, 2019.

Goyal, Y., Shalit, U., and Kim, B. Explaining classifiers with causal concept effect (cace). *CoRR*, abs/1907.07165, 2019.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 2018.

Gulshad, S. and Smeulders, A. Explaining with counter visual attributes and examples. In *Proceedings of the 2020 international conference on multimedia retrieval*, pp. 35–43, 2020.

Hernandez, E., Schwettmann, S., Bau, D., Bagashvili, T., Torralba, A., and Andreas, J. Natural language descriptions of deep features. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=NudBMY-tzDr.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *5th International Conference on Learning Representations, ICLR*, 2017.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *International conference on machine learning*, 2018.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. *International Conference on Machine Learning*, 2017.

Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., and Zhai, X. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

Laugel, T., Lesot, M.-J., Marsala, C., Renard, X., and Detyniecki, M. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. *International Joint Conference on Artificial Intelligence*, 2019.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Li, M., Zhao, Z., and Scheidegger, C. Visualizing neural networks with the grand tour. *Distill*, 5(3):e25, 2020.

Liu, M., Shi, J., Cao, K., Zhu, J., and Liu, S. Analyzing the training processes of deep generative models. *IEEE transactions on visualization and computer graphics*, 24 (1):77–87, 2017.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Michalowicz, J. V., Nichols, J. M., and Bucholtz, F. *Handbook of differential entropy*. Crc Press, 2013.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013b.

Nguyen, A. M., Yosinski, J., and Clune, J. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *Visualization for Deep Learning workshop at ICML*, 2016.

Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill*, 2(11):e7, 2017.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Papernot, N. and McDaniel, P. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

Park, H., Das, N., Duggal, R., Wright, A. P., Shaikh, O., Hohman, F., and Chau, D. H. P. Neurocartography: Scalable automatic visual summarization of concepts in deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 2021.

Pezzotti, N., Höllt, T., Van Gemert, J., Lelieveldt, B. P., Eisemann, E., and Vilanova, A. Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE transactions on visualization and computer graphics*, 24 (1):98–108, 2017.

Pühringer, M., Hinterreiter, A., and Streit, M. Instanceflow: Visualizing the evolution of classifier confusion at the instance level. In *2020 IEEE Visualization Conference (VIS)*, pp. 291–295. IEEE, 2020.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Rauber, P. E., Fadel, S. G., Falcao, A. X., and Telea, A. C. Visualizing the hidden activity of artificial neural networks. *IEEE transactions on visualization and computer graphics*, 23(1):101–110, 2016.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier. *ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.

Rice, L., Wong, E., and Kolter, Z. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Safarik, J., Jalowiczor, J., Gresak, E., and Rozhon, J. Genetic algorithm for automatic tuning of neural network hyperparameters. In *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything*, volume 10643, pp. 168–174. SPIE, 2018.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. *IEEE international conference on computer vision*, pp. 618–626, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Smilkov, D., Carter, S., Sculley, D., Viégas, F. B., and Wattenberg, M. Direct-manipulation visualization of deep networks. *arXiv preprint arXiv:1708.03788*, 2017.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. *IEEE conference on computer vision and pattern recognition*, 2016.

Targ, S., Almeida, D., and Lyman, K. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Yeh, C., Kim, B., Arik, S. Ö., Li, C., Pfister, T., and Ravikumar, P. On completeness-aware concept-based explanations in deep neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. Understanding neural networks through deep visualization. *International Conference on Machine Learning (ICML) Deep Learning Workshop*, 2015.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhang, Q., Wang, W., and Zhu, S.-C. Examining cnn representations with respect to dataset bias. *AAAI Conference on Artificial Intelligence*, 2018.

Zhong, W., Xie, C., Zhong, Y., Wang, Y., Xu, W., Cheng, S., and Mueller, K. Evolutionary visual analysis of deep neural networks. In *ICML Workshop on Visualization for Deep Learning*, pp. 9, 2017.

Zhou, B., Bau, D., Oliva, A., and Torralba, A. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018.

Zhou, Z., Li, K., Park, H., Dass, M., Wright, A., Das, N., and Chau, D. H. Neuromapper: In-browser visualizer for neural network training. *IEEE Visualization Conference (IEEE VIS)*, 2022.