CVPR 2019

LEARNING WORDS BY DRAWING IMAGES

D´ıdac Sur´ıs, Adria Recasens, David Bau, David Harwath,
James Glass, Antonio Torralba
Massachusetts Institute of Technology

ELİF ÇALIŞKAN 2016400183

30.10.2019

Introduction:

Learning Words by drawings is a paper with the subject of Vision and Language. The goal of this paper is to learn the relationship between spoken words and visual attributes such as shape, size and color. Since they work with human speech, the model should learn not only what a word means but also what a word is. There has been some researches regarding to this problem and they are clearly stated under Introduction title.

Motivation:

The paper "Jointly Discovering Visual Objects and Spoken Words from Raw Sensory Input"[1] shows that triplet loss can be used to learn visual objects via raw audio. But the topic of learning words for visual attributes such as colors and size has remained untouched.

Triplet loss is a loss function for artificial neural networks where a baseline input is compared to a positive input and a negative input. The distance from the baseline input to the positive input is minimized, and the distance from the baseline input to the negative input is maximized. This classification is used in triplet loss, it tries to reduce the distance/deviation between similar things and at the same time tries to increase the same between different things. This function is mainly used in this paper while clustering the image – audio matchmaps.

The paper called "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks"[0] was also mentioned in the Introduction part. This paper explains how GANs work and shows that with their work, it is possible to interactively manipulate objects in a scene such as ablating the image and restoring the background accordingly.

The purpose of GAN is to generate data from scratch -mostly images. GAN consists of two networks: generator and discriminator. Generator network is used for generating images using a normal or uniform distribution. Then this model is trained with real and generated images. In this training, GAN builds a discriminator to learn which attributes make an image real. This topic was considered to be known, but it can be hard to understand for uninformed readers like me.

As stated in "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning"[2] paper, CLEVR is used as a dataset to analyze a variety of modern visual reasoning systems, providing novel insights into their abilities and limitations. CLEVR is used in creating an audio description dataset of GAN generated images. In general, the previous work in literature has been explained superficially. An informed reader can understand the basis of this paper but as an unsophisticated reader I went through the annotated papers in order to understand the general concepts of GAN, triplet loss and CLEVR.

What remains to be solved:

In paper, there is no review part about their work and it is not emphasized that there are some unsolved parts. But in given examples, the visual attributes that the model has learned are shape, material, color and size. There are many attributes of an object, so these attributes can be enlarged. Also, this model can be used for facial parts of organisms such as cats and humans. This model can be used for face recognition by human speech if the research is carried forward.

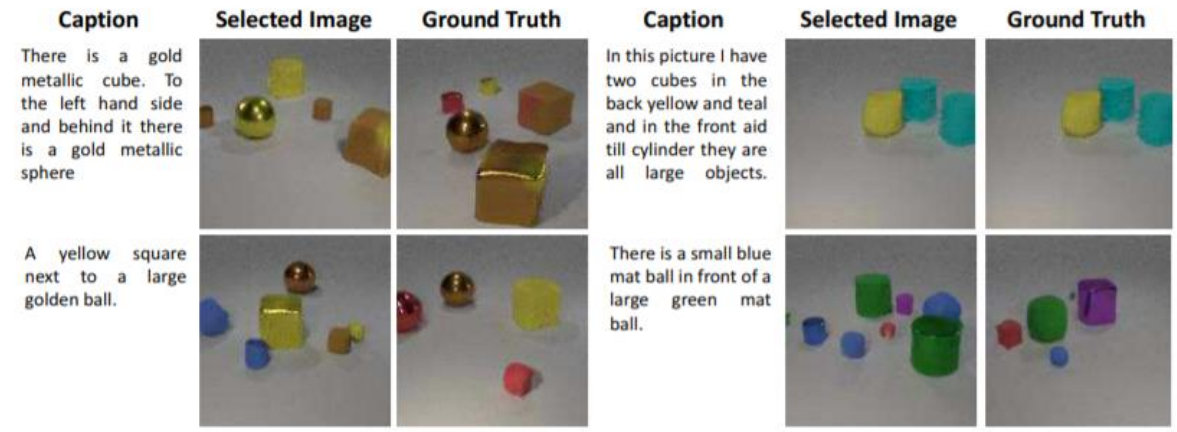

Figure 7: Examples of our system selecting images given a caption. Note that the retrieved image usually is closely related with the given description.

Contributions:

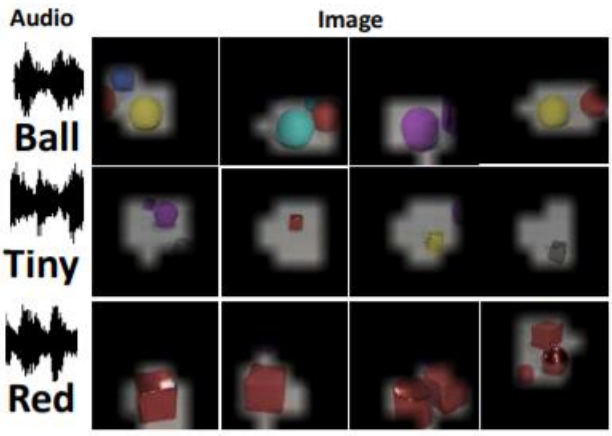

Figure 5: We show three examples of clusters learned by our model, represented by the images that mostly activate each cluster. We represent the audio-cluster with text for clarity, but all the learning is done in the audio domain. As shown, the system is able to learn color, shape and size.

The novelty of this paper is a learning framework that learns words by drawing images just like a child learns unnoticed details through drawing[3]. The novelty claim is valid because in Figure 5, there are three examples of clusters Ball, Tiny and Red. So we can say the system is able to learn color, shape and size. But since these visual attributes are very simple, the fact that with this model every shape, color and size can be classified is open to question. Also in Figure 7, there are some examples of this system selecting images given a caption. It is clear that the system proposes closely related images.

Model and assumptions:

Since this work is mainly based on previous researches, there are not many assumptions. The only assumption is that each verbal description provided by humans clearly represents the generated images.

Solution strategy:

The solution part starts at section 4.Editing Training Examples. For generator function g : R 100 $\longrightarrow$ I where I is the image domain. For every noise vector z $\in$ R 100 produces an image Iz = g(z). Then by fracturing the g(z) into gD and gE(z), gE(z) the output of the four initial convolutional layers is obtained. To ablate the pixel in (x,y) in dimension d, corresponding h4 value is assigned to 0. With ablating the object, new images are served as mismatched examples to audio captions which mention the object at pixel(x,y). This procedure was for editing the images randomly.

In the second part of training 4.2 Editing a specific attribute, rather than changing arbitrary attributes, specific ones are manipulated. For that, the segmentation function is used: s : I $\rightarrow$ {0, 1} c×w×h which outputs whether an image pixel contains an attribute of interest. Then, the most correlated filters are

removed in order to make a specific change. These s and g functions are nicely explained. Even though the layer concept was not very self-explanatory, in this paper the topic is covered concisely. After glancing at the GAN paper, these functions become clearer.

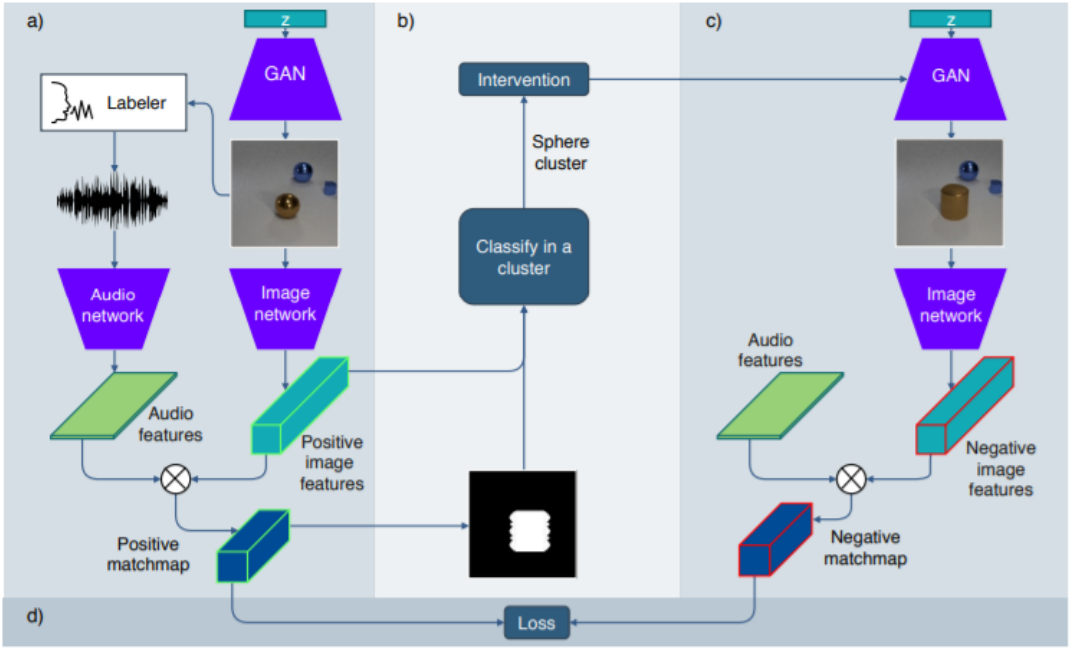

Figure 4: **Intervention schematic:** a) Basic model, where the original image and audio features are computed, as well as their matchmap. b) Clustering: highly activated image features are classified into a cluster, and an intervention is computed to generate an edited example. c) Generation of the edited example. The noise vector z is the same as in a). d) Triplet loss.

After training part, the solution continues with "5.Learning words by drawing images". At this part, the usage of the edited images is explained. Using the DaveNet model, two network consists of 512 dimensional representation called audio network and visual network. To obtain a score, dot product operation is done between audio network and visual network m(fI(I), fA(A)). This function generates a score for each point in time and space. Matchmap shows location of image and time of the related spoken word. The final similarity score f(I, A) is found by creating clusters based on maximum over image spatial dimensions and average over audio temporal dimension. The objective of f is to maximize the matchmap scores by using triplet loss: $L(I, A, In) = \max(f(In, A) - f(I, A) + \beta, 0)$. β is explained as an offset parameter but its meaning could have been clarified more. So, this function wants to increase the distance between the negative image while decreasing the distance from its corresponding image. Then, it is stated that L(I, A, An) is also should be minimized. But I could not understand clearly why it should be minimized. The clear definition of this function should be given just like the first equation. I am not convinced about why this function improves this model. Also In notation was used in equations but it is explained later as : In = gn(I, A) -output of negative generator.

General idea behind the solution is feeding the triplet loss in order to train the model. First, positive matchmaps are created using image and audio from human speech. Then, using these matchmaps and images, clusters are created based on their cooccurrence. After clusters are created, the concepts to be removed or edited are determined. Using the techniques learned from previous paper "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks"[0], these concepts are edited from images. The concepts learned by clusters can be seen in figure 5.

The general explanation of training procedure and used functions are clear and understandable. In general, the procedure is easy to follow, the mathematical part of the functions is not emphasized since the

previous paper [0] explains each function using mathematical expressions. But because of the definitions and given examples, it is obvious what each function does.

Analytical results:

For evaluation of this learning framework they compared different training methods such as: DaveNet, Hard Negatives, Random Edited Examples, Targeted Edited Examples, Hard Negatives and Random Edits, Hard Negatives and Targeted Edits. They listed the accuracy of the framework on each training method based on four different visual attributes: Shape, Material, Color and Size. Since the accuracy results are very high percentages and the experiments are made in various models, it is convincing that the framework can relate these attributes to their raw audio descriptions.

		Shape	Material	Color	Size	Mean
Human dataset	**DaveNet**	50.3	60.8	86.8	72.2	67.6
	Random Edits	52.0	48.9	87.8	91.3	70.0
	Target Edits	54.1	63.0	86.2	91.3	73.7
	Hard Neg	53.6	60.8	88.4	87.8	72.7
	HN+Random Edits	54.8	63.0	87.9	87.8	73.4
	HN+Target Edits	56.2	67.4	87.9	88.7	75.1
Synthetic dataset	**DaveNet**	72.6	63.3	51.1	98.0	71.2
	Random Edits	70.9	97.8	54.0	96.9	79.9
	Target Edits	69.3	97.5	57.9	95.4	80.1
	Hard Neg	75.6	91.3	62.2	97.6	81.7
	HN+Random Edits	73.3	94.5	70.5	95.1	83.3
	HN+Target Edits	77.7	96.9	66.6	97.1	84.6

Table 1: **Semantic accuracy**: We evaluate the ability of the different models to detect particular attributes in image. Given an audio with only the attribute, we ask the system to discriminate between images with and without the attribute.

Conclusions:

This paper represents a learning framework which can learn visual attributes and create relation with its corresponding audio by drawing. This framework uses negative -edited images for training the model. In order to make edited images, some methods learned in previous papers [0],[1] are used. These changes can be made on a random object in the image or can also be made on a salient concept of the image. For finding the salient concept, dot product of audio and visual network is computed. Then using triplet loss, clusters are made. After feeding the negative generator with the cluster to be edited, a negative image is created. Then this image and the original audio creates a negative matchmap and it is used in triplet loss again. This procedure creates training images and using this feedback mechanism, visual attributes such as shape, color, size and material can be learned by drawing. The experiments are made using various models and the mean accuracy of detecting particular attributes is very high.

References:

[0] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Visualizing and understanding generative adversarial networks. In International Conference on Learning Representations, 2019.

[1] D. Harwath, A. Recasens, D. Sur´ıs, G. Chuang, A. Torralba, and J. Glass. Jointly Discovering Visual Objects and Spoken Words from Raw Sensory Input. In European Conference on Computer Vision, 2018.

[2] J. Johnson, L. Fei-Fei, B. Hariharan, C. L. Zitnick, L. Van Der Maaten, and R. Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

[3] S. Butler, J. Gross, and H. Hayne. The effect of drawing on memory performance in young children. Developmental Psychology, 31:597–608, 07 1995.

Learning Words by Drawing Images

Dídac Surís * Adrià Recasens* David Bau David Harwath James Glass Antonio Torralba
Massachusetts Institute of Technology
{didac,recasens,davidbau,dharwath,glass,torralba}@csail.mit.edu
∗ indicates equal contribution

Abstract

We propose a framework for learning through drawing. Our goal is to learn the correspondence between spoken words and abstract visual attributes, from a dataset of spoken descriptions of images. Building upon recent findings that GAN representations can be manipulated to edit semantic concepts in the generated output, we propose a new method to use such GAN-generated images to train a model using a triplet loss. To apply the method, we develop Audio CLEVRGAN, a new dataset of audio descriptions of GAN-generated CLEVR images, and we describe a training procedure that creates a curriculum of GAN-generated images that focuses training on image pairs that differ in a specific, informative way. Training is done without additional supervision beyond the spoken captions and the GAN. We find that training that takes advantage of GAN-generated edited examples results in improvements in the model's ability to learn attributes compared to previous results. Our proposed learning framework also results in models that can associate spoken words with some abstract visual concepts such as color and size.

1. Introduction

Creation is an essential human learning process: simply drawing an object requires learning how to compose its parts, attributes, and relationships. Drawing helps children learn about details that would not otherwise be noticed [12]. In contrast, machine learning systems have not yet demonstrated an ability to learn through drawing. While Generative Adversarial Networks (GANs) have demonstrated dramatic success in learning to synthesize realistic images [32, 43], methods have not been developed for extracting other types of knowledge from a GAN.

In this work, our goal is to discover the correspondence between visual attributes and audio words from descriptions of images. Because we work with unannotated raw audio speech instead of text captions, our model must learn not only what a word means, but what a word *is* — our setting omits the structured supervision that would be provided by transcribed text. Previous work on this problem [24] has shown that a triplet loss can be used to learn a visually-grounded model of speech that can attend to relevant visual objects, but learning words for abstract visual attributes such as colors and shapes has been out of reach. In contrast, recent results suggest that GANs learn compositional visual concepts by learning to draw [9]. Yet a connection between language and the knowledge learned by GANs is missing.

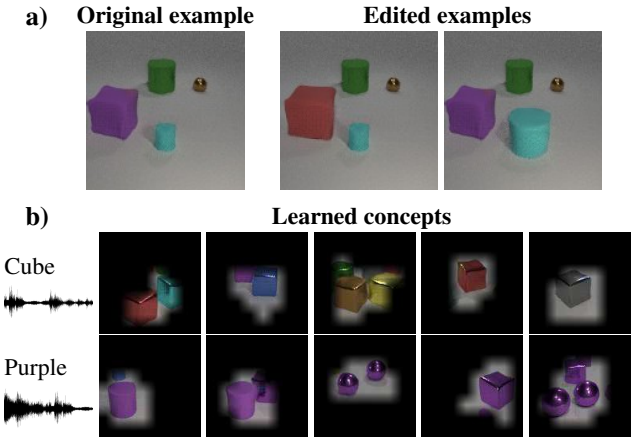

Figure 1: In this paper we propose a framework for learning through generated images. In a) two edited examples are compared to an original generated image at the left. In b) we show the concepts learned by our system when trained with edited examples.

We propose a training method that uses the knowledge learned by a GAN to generate a curriculum for training a visually-grounded model of spoken language [24]. Starting from a set of images with audio descriptions, we teach the model to learn about attributes by using the GAN to synthesize many realistic but meaningfully distinct training images (Figure 1a). The generation is done without any supervision beyond the original audio captions. These generated examples help the model focus on specific abstract visual attributes that correspond to audio words (Figure 1b).

We make the following contributions. First, we show how to use interventions to learn from a GAN. The key idea is to apply new results that show that a GAN can learn an internal disentangled representation where it is possible to control specific semantic aspects of the generated image [9]. This allows our system to generate artificial training examples which contain targeted differences that affect a small and controlled part of the image. Second, we apply this idea to multimodal training of an audio-image description model. To enable this, we introduce a new GAN-based dataset that includes both human and synthetic audio captions of GAN-generated CLEVR images.

2. Related work

Concept learning: There is an increasing interest in creating models that generalize by learning compositional concepts. It has been observed that deep networks that learn to classify scenes also learn to decompose those scenes into constituent object classes [11, 8, 45]. However, if appropriate concepts are not learned, it is easy for a model to be right for the wrong reasons: for example, answers to questions about images can be guessed without observing the order of the words in the question [46, 27]. It is argued in [39] that by monitoring and shaping input gradients, models can be trained to focus their attention on the right concepts.

A core challenge is to teach networks to extrapolate by applying learned rules to new situations, rather than only interpolating between similar inputs. To induce deep networks to learn to abstract reasoning, [7] trains models a dataset of compositional problems similar to human IQ tests. In [34], abstract attributes such as color, shape, or function are learned by modeling attributes as operators that relate objects to one another. In [3], an explicitly compositional architecture of re-usable neural modules was used for question answering.

Curriculum learning: The proposal that a training curriculum should be tailored to the evolving needs of the learner is foundational in machine learning [16, 10]. Curriculum learning remains an active area of current research, with both recent theoretical advances [21, 28] and practical applications [26, 19]. Our current work reformulates the curriculum problem by proposing that the training data can by synthesized by an expert teacher represented by a GAN.

CLEVR dataset: Our dataset is derived from the CLEVR dataset [29], a sythesized visual dataset consisting of scenes of simple objects with composable color, shape, and material attributes. CLEVR has been used to study abstract visual reasoning, question answering, and model interpretability [30, 41, 33]. We use CLEVR as a highly controlled visual domain in which compositional attributes are clear, and that a GAN can learn to draw well.

Generative Adversarial Networks: The quality and diversity of image generation results from GANs [20] has improved rapidly, from generating simple digits and faces [20], to synthesizing natural scene images [38, 14]. We use recent Progressive GAN [32] methods to generate high-resolution photorealistic images for creating training data. Furthermore, it has recently been found that GANs can add, remove, and modify objects in a scene by intervening in their internal representations directly [9]; we use that method to modify training examples.

Audio and Image: Real world objects are indicated not only by how they look, but how they sound. Recently proposed models that learn these correspondences can be used to perform tasks such as visually-guided audio source separation, or localizing source of a sound within an image or video [35, 44, 42, 17, 1, 18, 5]. Other works have also demonstrated the utility of audio-visual features for supervised classification tasks [4, 6], or predicting the sound made by an object [36, 37]. Another body of work has focused on learning words and other aspects of human language from spoken descriptions of visual images. This idea goes back to seminal work by [40], who introduced models that learned to associate images of everyday objects with phoneme sequences. More recently, [25, 22, 24] showed that models trained to associate visual images with spoken captions at the waveform level can implicitly discover a "dictionary" mapping between visual objects and spoken words, and [31] showed that the output of a visual object classifier could be used to train a keyword spotting system. Other works have investigated the emergence of different kinds of linguistic phenomena, such as sub-word units and phonemes, within similar models [13, 2, 15, 23].

3. Audio CLEVRGAN dataset

In this paper, we introduce a method to learn spoken words using GAN-generated images. To apply our method, we build a new dataset with spoken audio captions of GAN-generated images. Since our goal is to learn attributes, we train the generative system to synthesize the simple rendered images in the CLEVR dataset, in which the attributes and objects that appear in the images can be controlled. Although our method can be applied to natural images, the simplicity of the synthetic controlled environment helps human annotators provide detailed descriptions of attributes in the images, and it also makes it possible to obtain reliable attribute segmentation for evaluation.

To generate the images, we train a Progressive GAN [32] with the images in CLEVR dataset. We randomly sample the generative model to produce 20,000 images that are annotated by humans in Amazon Mechanical Turk using a similar interface as [25], where humans provide verbal descriptions of each image. We specifically ask annotators to mention the attributes and relations of different objects in the images. Examples of transcribed annotations are shown in Figure 2. We also generate a dataset of synthetic audio

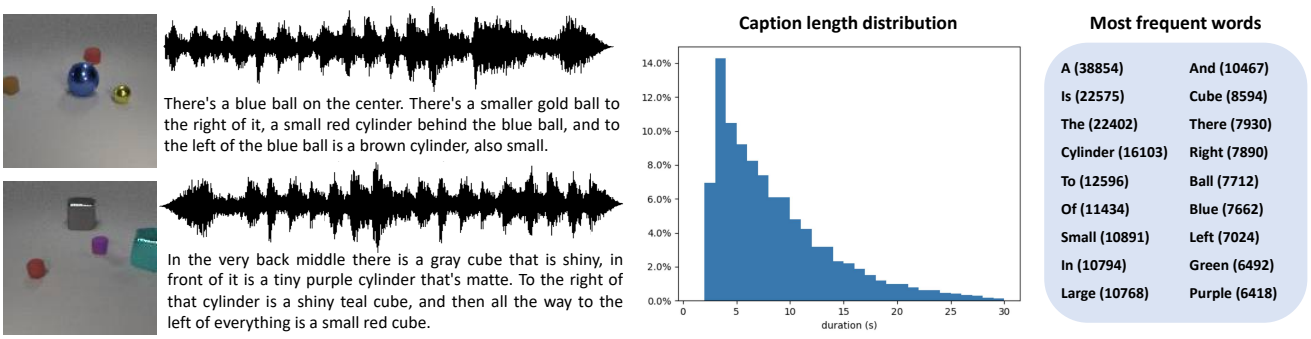

Figure 2: Examples of generated images and human annotated audios. In this figure, the transcriptions of the audio are shown instead of the audio, but no text transcriptions are used at any point during training or evaluation. We also provide some basic statistics of the Audio CLEVRGAN dataset.

captions, using the information from a previously trained attribute segmentation network. Each caption describes all attributes of all the objects in its associated image; we do this for 50,000 randomly sampled images from the GAN, including the 20,000 images annotated by humans.

4. Editing training examples

Learning attributes through spoken descriptions is a challenging task. Attribute words in a description are not observed in isolation and are usually tied to other attributes or nouns, which makes it difficult for a system to discover individual attribute words and isolate their meaning. To overcome this problem, we introduce the technique of generating targeted negatives by editing single visual attributes within images. Beginning with an image paired with a detailed description, we alter a single visual attribute in the image, after which the image will no longer match the original audio description. Such edited training examples will be used to guide the system to learn the correspondence between individual visual attributes and relevant audio words.

To edit visual attributes, we benefit from the rich internal representation learned by GANs [9]. These representations enable us to create edited versions of the original images where a single attribute is modified, as shown in Figure 3. In this section we describe how such edited training examples are generated. Then in Section 5 we use this method to learn a model that can isolate abstract attributes and match audio words with specific visual attributes.

4.1. Generating image edits

A trained GAN generator synthesizes images by processing a randomly sampled vector through a sequence of convolutions to produce a realistic image. It has been found that a GAN generator contains different sets of convolutional filters that specialize in generating different attributes and objects [9]. The activations of these convolutional filters can be modified to change, add, and remove objects in the output image. In this paper, we use this technique to modify certain attributes in particular objects.

Let $g : \mathbb{R}^{100} \longrightarrow \mathbb{I}$ denote a trained Progressive GAN generator for our dataset, where $\mathbb{I}$ is the image domain. Every noise vector $z \in \mathbb{R}^{100}$ produces an image $I_z = g(z)$. As in [9], we edit the image by manipulating the hidden representation in the fourth convolutional layer of the generator. We can write $g(z)$ as composition of two functions, $g(z) = g_D \circ g_E(z)$, where $h_4 = g_E(z)$ corresponds to the output of the four initial convolutional layers and g_D to the remaining layers. The representation at the fourth layer is a tensor $h_4 \in R^{512\times8\times8}$ in which some of the 512 dimensions correspond to the generation of certain objects or attributes. We can randomly ablate these values to randomly change attributes of objects.

Figure 3 shows the results of randomly ablating particular dimensions and locations of the representation h_4. To ablate the featuremap pixel (x, y) in dimension d, we set $h_4[d, x, y] = 0$. As expected, some attributes for the objects corresponding to the ablated location change. The new images can serve as mismatched examples as long as the object in the super pixel (x, y) is mentioned in the audio captions.

4.2. Editing a specific attribute

To further improve training, we wish to change specific attributes relevant to the audio description rather than arbitrary attributes. We do this by choosing the filters to ablate rather than ablating random filters.

Let $s : \mathbb{I} \rightarrow \{0, 1\}^{c\times w\times h}$ be a segmentation function that outputs a per-pixel binary classification predicting whether a image pixel contains an attribute of interest. By collecting statistics on a sample, we rank the filters of h_4 according to their correlation with s. Following the method in [9], we then ablate the specific filters of h_4 that are most highly correlated with s in order to remove the specific attribute identified by s from the generated image.

In [9], the segmentation functions s are pretrained to

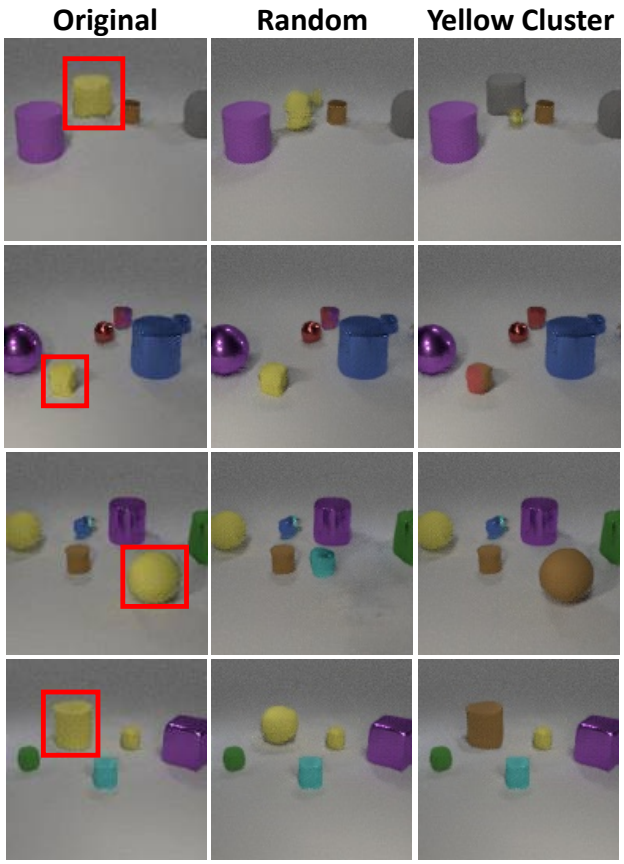

Figure 3: Examples of edited samples created using random editing and targeted interventions. In the left column, the original images with the target object in red. In the second column, randomly ablated units, applied to the same feature maps. Results range from distortions or complete change of the object (first and third rows), through useful semantic changes (fourth row), to barely noticeable changes (second row). In the last column, images generated by ablating the units corresponding to the yellow cluster. Ablating these units makes the yellow color change, as the cluster is representing this attribute.

identify ground-truth classes, but in our setting no ground truth segmentations are available. Instead, in Section 5.5 we shall derive guessed attribute segmentation functions from our model during training and use those guessed segmentation functions to select filters to ablate.

5. Learning words by drawing images

We now describe how we use edited training examples to improve the ability of an multimodal network to distinguish very similar concepts. We build upon previous work that learns concepts from spoken captions by using negative examples drawn from the training set [24]. We add training using edited GAN images to improve the model's ability to distinguish and isolate particular attributes. This is done in a multi-step training process that uses edited images that are successively more targeted as training proceeds.

The training process has the following steps. First, we train the basic system without any edited examples. Second, we use edited examples in which neurons are randomly ablated. This improves the internal representations of objects and attributes. Finally, we partition the space of audio-visual representation by clustering units according to co-occurrences. Each of these clusters correspond to different concepts present in the captions, such as colors, sizes, shapes, etc. We use these clusters to generate edited examples that are tailored to the mentioned concepts. The system is illustrated in Figure 4.

5.1. Architecture and triplet loss

We train the DaveNet model introduced in [24]. A schematic of the architecture is shown in Figure 4. DaveNet consists of two main networks: the audio network f_A and the visual network f_I. The audio network computes a 512 dimensional feature representation per each audio sample in a given window. Likewise, the image network generates a 512 dimensional representation per superpixel in the image.

To obtain a score, the two representations are combined through a dot product operation $m(f_I(I), f_A(A))$ which produces a map of scalar matching values for each point in space and time; we call this map a *matchmap*. Matchmap activations reveal the location and time of visual objects and spoken words that are related to each other in the model. We will later use the correspondence learned by the matchmap to guide the generation of edited examples by focusing edits on the most salient attributes and objects.

The final similarity score $f(I, A)$ between an image I and an audio description A is computed by aggregating matchmap activations, taking a max over image spatial dimensions and average over the audio temporal dimension.

The objective of f is to maximize the score of related pairs (I, A) given by the training set while minimizing the similarity of unrelated pairs (I_n, A). Following the method of [24], we train f using the triplet loss:

$$L(I, A, I_n) = \max(f(I_n, A) - f(I, A) + \beta, 0) \qquad (1)$$

where β is an offset parameter. Analogously, we also minimize $L(I, A, A_n)$. Both losses are combined in training.

5.2. Using edited images as negative examples

The selection of negative examples I_n has long been an important topic in computer vision. Previous work [24] proposed using random samples or mismatched samples that the network classifies closest to the threshold. These methods assume a closed set of images from which to choose, but none entertain the possibility of *creating* mismatched

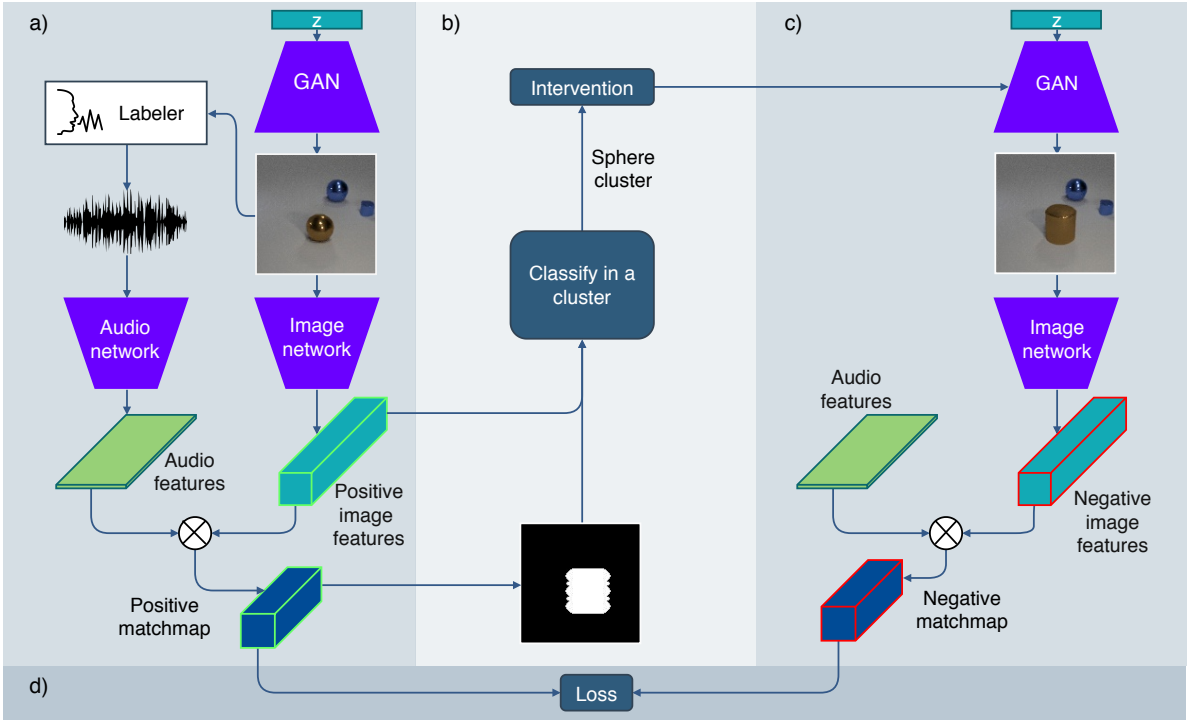

Figure 4: **Intervention schematic:** a) Basic model, where the original image and audio features are computed, as well as their matchmap. b) Clustering: highly activated image features are classified into a cluster, and an intervention is computed to generate an edited example. c) Generation of the edited example. The noise vector z is the same as in a). d) Triplet loss.

samples to aid learning. We use interventions in the GAN to generate ideal counterexamples to pair with each positive image. The edited negative examples will improve performance on the most confusing cases.

We will use g_n to denote our negative sample generation algorithm. Given an image I and an audio A, it will create an edited negative sample $I_n = g_n(I, A)$, that will only differ from I in a small set of characteristics. Using the technique of Section 4.1, the generator g_n will generate I_n using the same representation h_4 that was used to generate I, but modified by ablating some of the neurons in the location of the edited content. One key question remains: how do we select which neurons to ablate to generate the best possible edited example? The following section describes a multi-step training process that determines these units to get increasingly more targeted edits as training proceeds.

5.3. Model initialization

In the first step of the process, the model is pretrained using randomly sampled negatives as in [24]. The original triplet loss is used, and edited examples are not synthesized. This initialization bootstraps the model so that the matchmap can detect regions of the image that are salient to the description. This pretrained model can locate objects, but it cannot fully disentangle specific object attributes.

5.4. Randomly edited examples

The next step is to train the network with randomly edited examples. Each edited image is generated by using the matchmap $m(f_I(I), f_A(A))$ to identify the most salient location in a positive image-caption pair, and then randomly ablating GAN feature channels at that location in the image. Each channel is ablated with probability $p = 0.2$, which is increased until the edited I_n differs from the original I.

This random ablation strategy generates a wide variety of edited examples as seen in Figure 3. While some of the modified images are informative negatives that falsify a single word in the caption, others may be too similar to the positive image to correspond to any caption change; and others may be different enough to correspond to differences in many words. While this mix of edited examples is more informative than random negative images chosen from the input batch, we perform yet another training phase to generate higher-quality negatives.

5.5. Clustering

The ideal edited example would differ from an original image by just one attribute of one object; a minimal change would match the original caption in all words except for one. However, as we are dealing with a continuous audio

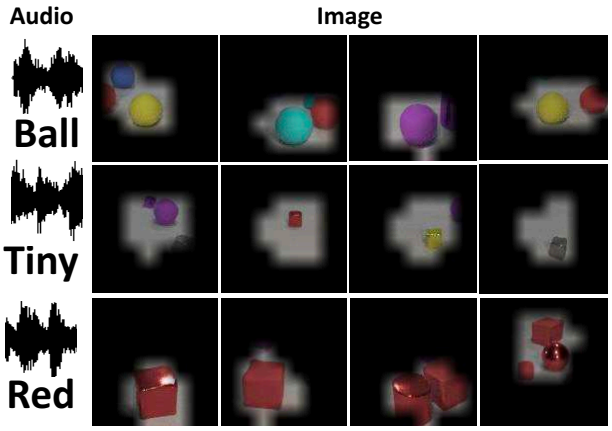

Figure 5: We show three examples of clusters learned by our model, represented by the images that mostly activate each cluster. We represent the audio-cluster with text for clarity, but all the learning is done in the audio domain. As shown, the system is able to learn color, shape and size.

signal, word boundaries are unknown, and such minimal concepts must be learned. In this stage, we create a set of word-like concepts by clustering the learned features of f. This grouping enables us to choose how to intervene the GAN in order to change a single descriptive word.

To build the concept clusters, we process the full training set through our audio-visual model and observe the audio-visual features that activate in each training pair. We binarize these by considering only activations in the top 1% percentile. Finally, we compute a co-occurrence matrix of the binarized features to measure how much every pair of neurons co-activate. This enables us to partition the neuron space using a dendrogram, grouping units with high co-occurrence. This clustering in the unit space induces a semantic clustering in the shared embedding space of the matchmap. Figure 5 shows some examples of clusters. The image clusters are coherent and usually represent a concept in the image space, while the audio usually represents one or a few spoken words with the same meaning. We refer to a unit cluster as w_k.

Learning how remove a concept from an image: As described in Section 4.2, a segmentation function $s(I)$ that locates a concept in an image can be used to identify GAN units that generate that concept in an image [9]. Although we do not have ground truth segmentation for abstract visual attributes, we can use a cluster-inferred segmentation to achieve the same effect. We define a binary segmentation function $s(I|w_k)$ to select pixel locations that activate w_k units of the matchmap representation $f_I(I)$. We then apply the procedure described in 4.2 to identify the units of the generator that are responsible for generating the visual concept corresponding to the cluster w_k.

Finally, we generate *targeted edited examples* that make changes that affect cluster w_k by ablating the GAN units associated with w_k. This modifies the image by changing aspects of the image that are guessed to correspond to one concept: this avoids random edited examples that are too similar or too different. Note that for this method to be effective, we must cluster units that already carry some information about disentangled concepts. Such units can be initially learned by training with random edited examples.

5.6. Training with targeted edited examples

To create the edited mismatched example, we use the following procedure as presented in Figure 4:

1. Given a pair of image and audio (I_i, A_i), we compute $f_I(I_i)$ and $f_A(A_i)$.
2. We identify the feature embedding of the most salient visual concept $w_i = f_I(I, i)^{(x,y)}$ where $x, y, t = \text{argmax}_{x,y,t} \left(f_I(I_i)^{(x,y)} \cdot f_A(A_i)^{(t)}\right)$.
3. We compute the similarity between w_i and each cluster w_k. We randomly draw a cluster w_k with probability in proportion to this score.
4. Using the intervention procedure, we ablate the GAN neurons associated with w_k to generate an edited example for that particular attribute: $I_i^n = g_n(I_i)$.
5. Then we use $f_I(I_i^n)$ as a negative and train the model using backpropagation.

6. Experiments

In this section, we evaluate the proposed learning framework in various experimental settings. For all our experiments, we use the DaveNet network with the same configuration as in [24]. It consists of an image and an audio branch, the two of them fully convolutional. For human annotated data, we increase the depth of the audio model adding three extra convolutional layers at the end. For the synthetic dataset we maintain the original size. To train the Progressive GAN, we used the same parameters as in [32].

6.1. Synthetic dataset creation

To provide a better picture of the different possibilities of our model, we created synthetic descriptions for the GAN generated images. To do so, we trained a segmenter in the original CLEVR dataset, which contains ground truth information about attributes and objects. Using these segmentations, we created one description per image, in a similar style as human captions. The description includes all the objects with their corresponding attributes, as well as the spatial relation between them.

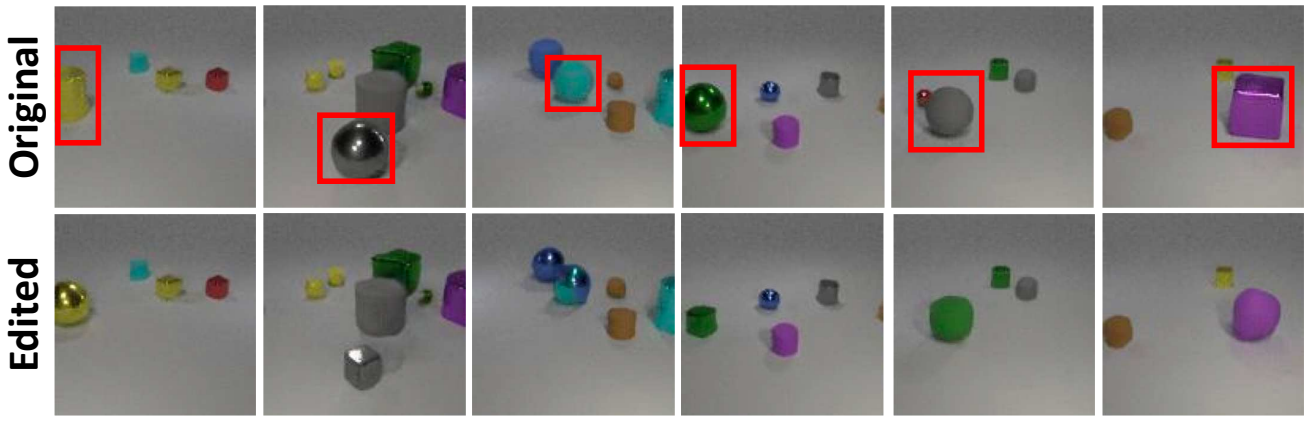

Figure 6: In this figure we show multiple examples of edited images using our targeted algorithm. Note that the system is able to modify particular attributes of the object.

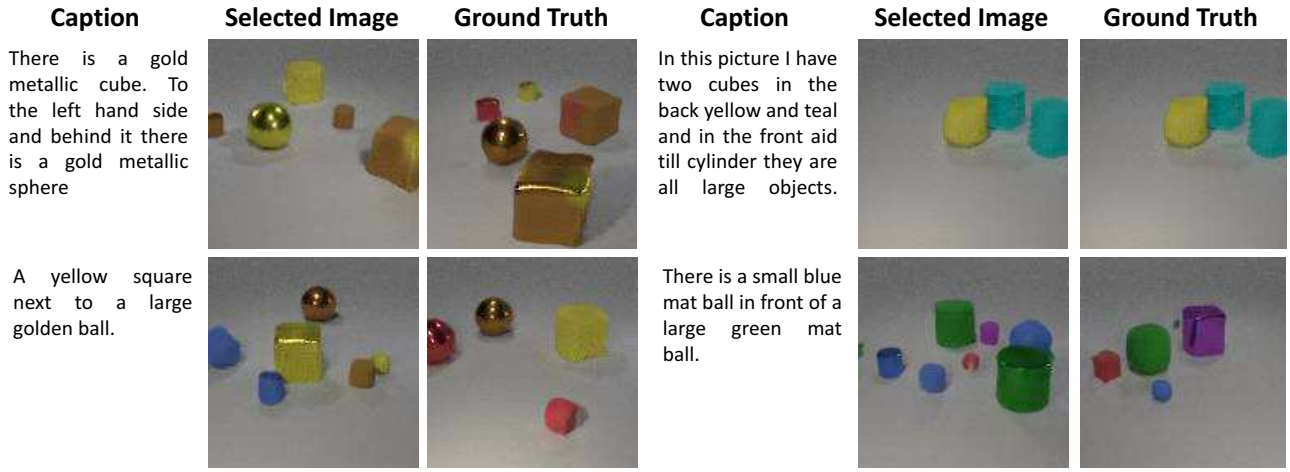

Figure 7: Examples of our system selecting images given a caption. Note that the retrieved image usually is closely related with the given description.

6.2. Evaluation setting

To understand if a concept has been learned by the system, it is necessary to test it in isolation from other concepts. Neural networks can learn to create global representations, but fail at representing specific attributes. In this section we propose a *semantic test*, in which we test the models to recognize isolated attributes. For each attribute, we produce pairs of images, one containing the attribute and another without the attribute. We then create an input for the audio network containing the isolated attribute to be evaluated in the form of a spoken word. We can compute the accuracy of the system on selecting the image with the attribute against the image without the attribute. In addition to the semantic test, we also show the recall on random negatives, where 500 image-audio pairs of a held-out test set are passed through the network, and the retrieval recalls from audio to image and from image to audio are computed.

6.3. Methods

For evaluation, we compare many different training methods. **DaveNet**: The training procedure in [24], where random negatives are used. **Hard Negatives**: The negative image and audio are selected as the sample in the minibach with highest loss. **Random Edited Examples**: The examples produced by random ablation in of the hidden representation in the GAN. **Targeted Edited Examples**: The examples produced according to the semantics of the object intervened. **Hard Negatives + Random Edits**: We combine the random edited examples with the hard negative loss. In training, we use the hardest negative of both methods. **Hard Negatives + Targeted Edits**: We combine the targeted edited examples with the hard negative loss.

		Shape	Material	Color	Size	Mean
Human dataset	**DaveNet**	50.3	60.8	86.8	72.2	67.6
	Random Edits	52.0	48.9	87.8	91.3	70.0
	Target Edits	54.1	63.0	86.2	91.3	73.7
	Hard Neg	53.6	60.8	88.4	87.8	72.7
	HN+Random Edits	54.8	63.0	87.9	87.8	73.4
	HN+Target Edits	56.2	67.4	87.9	88.7	75.1
Synthetic dataset	**DaveNet**	72.6	63.3	51.1	98.0	71.2
	Random Edits	70.9	97.8	54.0	96.9	79.9
	Target Edits	69.3	97.5	57.9	95.4	80.1
	Hard Neg	75.6	91.3	62.2	97.6	81.7
	HN+Random Edits	73.3	94.5	70.5	95.1	83.3
	HN+Target Edits	77.7	96.9	66.6	97.1	84.6

Table 1: **Semantic accuracy**: We evaluate the ability of the different models to detect particular attributes in image. Given an audio with only the attribute, we ask the system to discriminate between images with and without the attribute.

	Human Dataset			**Synthetic Dataset**		
	R@1	**R@5**	**R@10**	**R@1**	**R@5**	**R@10**
DaveNet	8.4	26.3	38.5	14.9	43.7	62.2
Random Edits	12.5	33.8	49.8	60.6	89.0	95.1
Targeted Edits	14.1	37.2	52.2	75.1	95.5	98.5
Hard Neg	20.5	45.1	60.7	73.4	94.6	97.6
HN+Random	19.3	48.3	63.0	94.8	99.7	99.9
HN+Targeted	20.3	49.3	61.9	93.4	99.6	99.9

Table 2: **Results in the Audio CLEVRGAN dataset**: Recall results (in %) for the two datasets, for the different methods, showing that more refined interventions get better results. Recall in the random test is over 500 samples.

In training, we select the hardest negative of both methods. Note that the Random Edit model has been trained initializing with DaveNet, and the Targeted Edit model has been trained initializing with the Random Edit model. The same procedure is used for the models with Hard Negatives.

6.4. Results

In Table 1, we report the accuracy of our method and the baselines for the semantic test, both in the human captioned dataset and the synthetic generated dataset. We break down the results in the different attributes in our dataset. As expected, the basic DaveNet model performs poorly in this test, suggesting that the system is not able to learn particular isolated concepts. Furthermore, the models using targeted edits have a better ability on predicting particular attributes, which reinforces the idea that using edited examples for training increases the model understanding of isolated attributes. Finally, human models focus more its attention on discriminating color as they are more mentioned in the audio captions. However, when using the synthetic captions, where attributes are evenly distributed, performance drops on discriminating color but increases for the other attributes.

In Table 2 we report the average of the caption to image and image to caption recall for all the models in 500 images of the held out test set. First, the usage of edited images already improves performance over the DaveNet baseline, suggesting the edited examples positively contribute to the learning process. Furthermore, when mixed with the hard negative loss, the models increase significantly its recall ability. Note that performances in the synthetic dataset are consistently higher, as descriptions are more informative. In Figure 6, we show our system's ability to edit images. It is able to modify different attributes of the objects such as shape or color. We found that our system successfully changes the caption content in 88% of the edits. Finally, in Figure 7, we show some examples of retrieved images using our method on the held out test set. Our system does retrieve images which largely match the caption, sometimes only missing one particular object or attribute.

6.5. Generalizing to real images

Discriminating concepts and attributes is useful when it can be applied to the original images, not just in the GAN-generated domain. To test how well the knowledge transfers to the original non-GAN-generated CLEVR images, we created a test dataset consisting of 1000 original CLEVR images with their corresponding edited examples (changing only one attribute of one object). Given a synthetic caption, the system must choose between the positive and the negative (chance being 50%). A model trained on original CLEVR images, with a regular DaveNet without edited examples, has an accuracy of 54%, showing that a regular model struggles to learn specific attributes. A model trained on GAN-generated images with edited examples also generated by the GAN has an accuracy of 59%, even when not trained on original images. This suggests that our method can be transferred to the original images domain. We expect these gains to improve as GAN algorithms improve. Having access to the CLEVR renderer, we can synthesize edited examples programmatically. Training a system with these edited images we get an upper bound accuracy of 89% on this test.

7. Conclusions

We presented a learning framework that learns words by drawing images. We take advantage of the fact that generative models have already learned many concepts about the visual word in order to edit images. These edited images are used to train an audio-visual system that can localize words in an image. We showed how the model itself can be used to improve the edited images. Finally, we evaluated the proposed methods in the Audio CLEVRGAN dataset.

References

[1] T. Afouras, J. S. Chung, and A. Zisserman. The conversation: Deep audio-visual speech enhancement. In *INTERSPEECH*, 2018.

[2] A. Alishahi, M. Barking, and G. Chrupala. Encoding of phonology in a recurrent neural model of grounded speech. In *CoNLL*, 2017.

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1545–1554, San Diego, California, June 2016. Association for Computational Linguistics.

[4] R. Arandjelovic and A. Zisserman. Look, listen, and learn. In *ICCV*, 2017.

[5] R. Arandjelovic and A. Zisserman. Objects that sound. In *ECCV*, 2018.

[6] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems 29*, pages 892–900. 2016.

[7] D. G. T. Barrett, F. Hill, A. Santoro, A. S. Morcos, and T. Lillicrap. Measuring abstract reasoning in neural networks. In J. Dy and A. Krause, editors, *Proc. 35th Int. Conf. Mach. Learn.*, pages 511–520, Stockholm, Sweden, 2018. PMLR.

[8] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *CVPR*, 2017.

[9] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[11] A. T. Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva. Object detectors emerge in deep scene CNNs. In *Int. Conf. Learn. Represent.*, 2015.

[12] S. Butler, J. Gross, and H. Hayne. The effect of drawing on memory performance in young children. *Developmental Psychology*, 31:597–608, 07 1995.

[13] G. Chrupala, L. Gelderloos, and A. Alishahi. Representations of language in a model of visually grounded speech signal. In *ACL*, 2017.

[14] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[15] J. Drexler and J. Glass. Analysis of audio-visual features for unsupervised speech recognition. In *Grounded Language Understanding Workshop*, 2017.

[16] J. L. Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.

[17] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *SIGGRAPH*, 37:112:1–112:11, 2018.

[18] R. Gao, R. Feris, and K. Grauman. Learning to separate object sounds by watching unlabeled video. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[19] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260, 2016.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[21] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*, 2017.

[22] D. Harwath and J. Glass. Learning word-like units from joint audio-visual analysis. In *ACL*, 2017.

[23] D. Harwath and J. Glass. Towards visually grounded sub-word unit discovery. In *ICASSP*, 2019.

[24] D. Harwath, A. Recasens, D. Surís, G. Chuang, A. Torralba, and J. Glass. Jointly Discovering Visual Objects and Spoken Words from Raw Sensory Input. In *European Conference on Computer Vision*, 2018.

[25] D. Harwath, A. Torralba, and J. R. Glass. Unsupervised learning of spoken language with visual context. In *NIPS*, 2016.

[26] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, page 6, 2017.

[27] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. In *European conference on computer vision*, pages 727–739. Springer, 2016.

[28] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann. Self-paced curriculum learning. In *AAAI*, volume 2, page 6, 2015.

[29] J. Johnson, L. Fei-Fei, B. Hariharan, C. L. Zitnick, L. Van Der Maaten, and R. Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] J. Johnson, B. Hariharan, L. Van Der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Inferring and Executing Programs for Visual Reasoning. In *International Conference on Computer Vision (ICCV)*, 2017.

[31] H. Kamper, S. Settle, G. Shakhnarovich, and K. Livescu. Visually grounded learning of keyword prediction from untranscribed speech. In *INTERSPEECH*, 2017.

[32] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[33] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar. Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning. *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[34] T. Nagarajan and K. Grauman. Attributes as Operators. *European Conference on Computer Vision*, 2018.

[35] A. Owens and A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *European Conference on Computer Vision (ECCV)*, September 2018.

[36] A. Owens, P. Isola, J. H. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2405–2413, 2016.

[37] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. *Ambient Sound Provides Supervision for Visual Learning*, pages 801–816. 2016.

[38] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[39] A. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2662–2670, 2017.

[40] D. Roy and A. Pentland. Learning words from sights and sounds: a computational model. *Cognitive Science*, 26:113–146, 2002.

[41] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Conference on Neural Information Processing Systems (NIPS)*, 2017.

[42] A. Senocak, T.-H. Oh, J. Kim, M.-H. Yang, and I. So Kweon. Learning to localize sound source in visual scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[43] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[44] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba. The sound of pixels. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[45] B. Zhou, D. Bau, A. Oliva, and A. Torralba. Interpreting Deep Visual Representations via Network Dissection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (*, 2017.

[46] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, 2015.