AFRL-RY-WP-TR-2020-0212

LIFELONG VISUAL EPISODIC MEMORY

David Jacobs and Tom Goldstein
University of Maryland

Ronen Basri
The Weizmann Institute of Science

SEPTEMBER 2020
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with The Under Secretary of Defense memorandum dated 24 May 2010 and AFRL/DSO policy clarification email dated 13 January 2020. This report is available to the general public, including foreign nationals.

Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RY-WP-TR-2020-0212 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

OLGA L. MENDOZA-SCHROCK
Program Manager
Decision Sciences Branch
Multi-Domain Sensing Autonomy Division

*//Signature//

OLGA L. MENDOZA-SCHROCK, Chief
Decision Sciences Branch
Multi-Domain Sensing Autonomy Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
September 2020	Final	16 March 2018 – 30 June 2019

4. TITLE AND SUBTITLE
LIFELONG VISUAL EPISODIC MEMORY

5a. CONTRACT NUMBER
FA8650-18-2-7833

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61101E

6. AUTHOR(S)
David Jacobs and Tom Goldstein (University of Maryland)
Ronen Basri (The Weizmann Institute of Science)

5d. PROJECT NUMBER
1000

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
Y1SE

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Maryland
College Park, MD, 20742

Weizmann Institute of Science
234 Herzl St. PO Box 26,
Rehovot 7610001, Israel

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH
45433-7320
Air Force Materiel Command
United States Air Force

Defense Advanced Research
Projects Agency
DARPA/MTO
675 North Randolph Street
Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYAT

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TR-2020-0212

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with The Under Secretary of Defense memorandum dated 24 May 2010 and AFRL/DSO policy clarification email dated 13 January 2020. This material is based on research sponsored by the Air Force Research Lab (AFRL) and the Defense Advanced Research Agency (DARPA) under agreement number FA8650-18-2-7833. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air force Research Laboratory (AFRL) and Defense Advanced Research Agency (DARPA) or the U.S. Government. Report contains color.

14. ABSTRACT
This report describes work on Lifelong Learning that develops methods for using Generative Adversarial Networks (GANs) to represent the probability of images. This problem is integral to developing the ability to detect and characterize domain shift. It also explores causality in video. Causal reasoning may play a key role in the ability of an agent to adapt to changing environments. If one understands what components of the environment lead to a particular outcome, one can determine whether changes to the environment affect these causal factors and should affect expected outcomes. It also explores semantic segmentation in the presence of domain shift.

15. SUBJECT TERMS
lifelong learning, domain shift, generative adversarial networks, semantic segmentation, causality, video understanding

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Asif Mehmood
Unclassified	Unclassified	Unclassified	SAR	29	19b. TELEPHONE NUMBER *(Include Area Code)* N/A

1. Introduction

Our work on L2M has spanned a number of topics. In the beginning of the grant, we focused on developing methods for using Generative Adversarial Networks (GANs) to represent the probability of images in a specific domain. Solving this problem is integral to developing a fine-grained ability to detect and characterize domain shift. We also attacked a more concrete problem of performing semantic segmentation in the presence of domain shift. As a third thrust of our work, we have explored the problem of understanding causality in video. We conjecture that causal reasoning plays a key role in the ability of an agent to adapt to changing environments. If one understands what components of the environment lead to a particular outcome, one can determine whether changes to the environment affect these causal factors, and should affect expected outcomes. In this report we describe each of these three thrusts.

2. Representing Probability Distributions of Images

Researchers have long sought to estimate the probability density functions (PDFs) of images. The resulting generative models can be used in image synthesis, outlier detection, image restoration, and in classification. There have been some impressive successes, including building generative models of textures for texture synthesis, and using low-level statistical models for image denoising. However, building accurate densities for full, complex images remains challenging.

Recently there has been a flurry of activity in building deep generative models of complex images, including the use of generative adversarial networks (GANs, Goodfellow, 2014) to generate stunningly realistic complex images. While some deep models focus explicitly on building probability densities of images, the emphasis has been on GANs that generate the most realistic imagery. Implicitly, though, these GANs also encode probability densities. In our work we explore whether these implicit densities capture the intuition of a probable image. We show that in some sense the answer is ``no''. But we suggest that by computing PDFs over latent representations of images, we can do better.

We first propose some methods for extracting probability densities from GANs. It is well known that when a bijective function maps one density to another, the relationship between the two densities can be understood using the determinant of the Jacobian of the function. GANs are not bijective and map a low-dimensional latent space to a high-dimensional image space. In this case, we modify the standard formula so that we can extract the probability density value of an image given its latent representation. This allows us to compute densities of images generated by the GAN, which we then use to train a regressor that computes densities of arbitrary images.

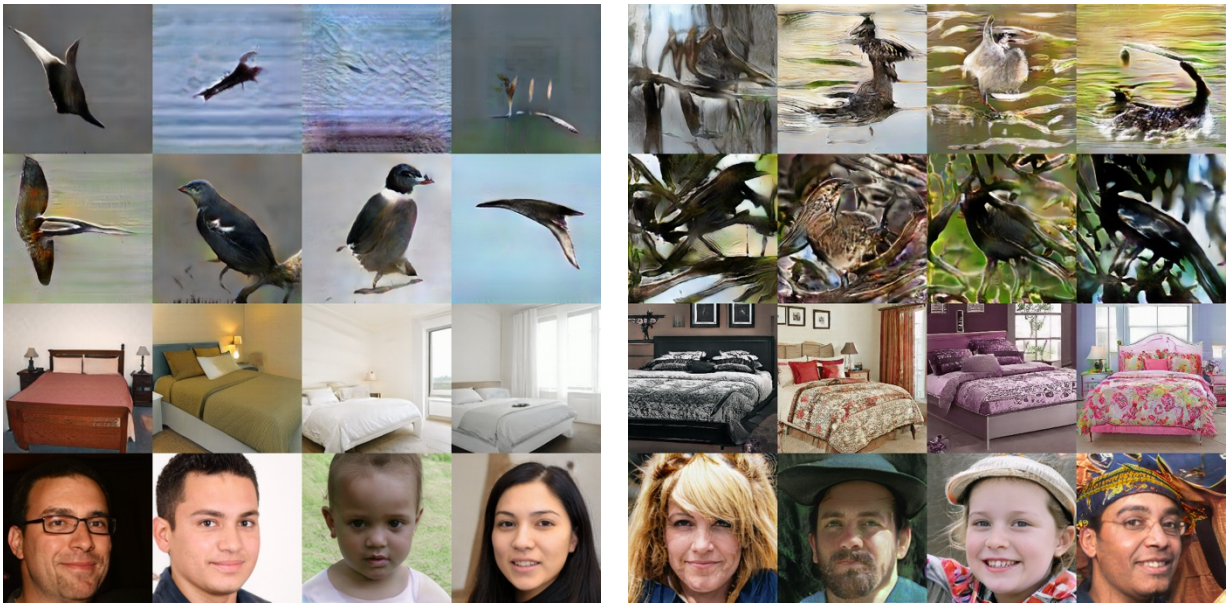

Figure 1 The images with the highest and lowest densities among 100 samples from a StackGAN (top two rows) and a StyleGAN (bottom two rows). **Left:** Images with highest density. **Right:** Images with lowest density. **Top row:** Samples from StackGAN trained on the CUB-200 dataset (Welinder et al., 2010), conditioned on the caption ``A bird with a very long wing span and a long pointed beak.'' **Second row:** Samples from StackGAN conditioned on the caption ``This bird has a white eye with a red round shaped beak.'' **Third row:** Samples from a StyleGAN model pretrained on the LSUN Bedroom dataset (Yu et al., 2015). **Bottom row:** Samples from a StyleGAN model pretrained on the Flickr-Faces-HQ dataset (Karras et al., 2018).

We perform sanity checks to ensure that GANs do indeed produce reasonable densities on images. We show that GANs produce similar densities for training images and for held out test images from the same distribution. We also show that when we compute the density of either real or generated images, the most likely (highest density value) images are of low complexity, and the least likely images are of high complexity. An example of this last result is shown in Figure 1, which displays the images with highest and lowest densities among samples generated by a StackGAN (Zhang et al., 2017) and a StyleGAN (Karras et al., 2018). The StackGAN images are conditioned on two different captions, and the StyleGAN images are from models trained on two different datasets.

Unfortunately, we also show that probability densities learned on images are difficult to interpret and have unintuitive behaviors. The strong influence of visual complexity on the learned PDF causes irrelevant background details to dominate the shape of the distribution; we see that the most likely images tend to contain small objects with large, simple backgrounds, while images with complex backgrounds are deemed unlikely despite being otherwise sensible. For example, for a GAN trained on MNIST, all of the most likely digits are 1, despite each type of digit occurring in equal proportion in the training set. If we exclude 1s from the training data and then compute the densities of all MNIST digits under this altered distribution, the most likely digits are *still* 1s, even though the GAN never saw them during training. In fact, even if we train a GAN on CIFAR images of real objects, the GAN will produce higher densities for MNIST images

of 1s than for most of the CIFAR images. We investigate these strange properties of density functions in detail and explore reasons for this lack of interpretability.

We propose to mitigate this problem by doing probability density estimation on the latent representations of the images, rather than their pixel representations. With this approach we obtain probability distributions with inliers and outliers that seem to coincide more closely with our intuition.

In parallel to our work, Nalisnick et al. (2018) also addresses the interpretability of density functions over images, claiming that seemingly uninterpretable density estimates result from inaccurate estimation on out-of-sample images. Our thesis is different, as we argue that density estimation is often accurate even for unusual images, but the true underlying density function (even if known exactly) is fundamentally difficult to interpret.

2.1 Background

There are many classical models for density estimation in low-dimensional spaces. Non-parametric methods such as Kernel density estimation (i.e., Parzen windows, Parzen, 1962; Heidenreich et al., 2013) can model simple distributions with light tails, and nearest-neighbor classifiers (eg., Boiman et al., 2008) implicitly use this representation. Directed graphical models (eg., Chow-Liu trees and related models (Chow and Liu, 1968) have also been used for classification (Mattar and Learned-Miller, 2006). However, these models do not scale up to the complexity or dimensionality of image distributions.

There is a long history of approximating the PDFs of images using simple statistical models. These approaches succeed at estimating some low-dimensional marginal distribution of the true image density. Modeling the complete, high-dimensional distribution of complex images is a substantially more difficult problem. For example, Olshausen and Field (1996) models the low-level statistics of natural images. Portilla and Simoncelli (2003) uses conditional models on the wavelet coefficients of images and shows that these models can improve image denoising. Roth and Black (2005) learns and applies image priors based on Fields of Experts. Markov models have also been used to synthesize textures with impressive realism (De Bonet and Viola, 1998; Efros and Leung, 1999).

Neural networks have been used to build generative models of images. Park (2016) and Timofte et al. (2012) do so assuming independence of pixels or patches. Restricted Boltzmann Machines (Smolensky, 1986) and Deep Boltzmann machines (Salakhutdinov & Larochelle, 2010) also model image densities. However, these methods suffer from complex training and sampling procedures due to mean field inference and expensive Markov Chain Monte Carlo methods (Salimans et al., 2015). In another approach, Variational Autoencoders (Kingma & Welling, 2013) simultaneously learn a generative model and an approximate inference, and offer a powerful approach to modeling image

densities. However, they tend to produce blurry samples and are limited in application to low-dimensional deep representations.

Recently, GANs (Goodfellow et al., 2014) have presented a powerful new way of building generative models of images with remarkably realistic results (Brock et al., 2018). Generative adversarial networks are neural network models trained adversarially to learn a data distribution. They consist of a generator G_θ: R^n -> R^m and a discriminator D_ϕ: R^m -> R, where n is the dimension of a latent space with probability distribution P_z and m is the dimension of the data distribution P_d, which is equal to width x height x #colors in the case of images. In the original GAN, the discriminator produces a probability estimate as output, and the GAN is trained to reach a saddle point via the learning objective

$$\min_\theta \max_\phi \mathbb{E}_{x\sim P_d}[\log D_\phi(x)] + \mathbb{E}_{z\sim P_z}[\log(1 - D_\phi(G_\theta(z))], \quad (1)$$

which incentivizes the generator to produce samples that the discriminator classifies as likely to be real, and the discriminator to assign high probability values to real points and low values to fake points. Unfortunately, GANs don't produce explicit density models -- the GAN is capable of sampling the density, but not evaluating the density function directly.

A major limitation of GANs is that they are not invertible. So, given an image, one does not have access to its latent representation, which could be used to calculate the image's density value. To overcome this problem, Real Non-Volume-Preserving transformations (Real NVP; Dinh et al., 2016) learn an invertible transformation from the latent space to images. This yields an explicit probability distribution in which exact density values can be computed. Real NVP can be trained using either maximum likelihood methods or adversarial methods, or a combination of both, as in FlowGAN (Grover et al., 2017). Both of these models have proven effective at generating high-quality images. (See also: Dinh et al., 2014; Papamakarios et al. 2017).

In our work we choose to focus on the use of non-invertible GANs to estimate image density. One issue with invertible GANs is that the latent space must be of the same dimension as the image space, which becomes problematic for large, high-dimensional images. Also, non-invertible GANs currently produce the highest quality images, suggesting that they implicitly represent the most accurate probability distributions. Furthermore, non-invertible GANs use simpler network architectures and training procedures. The standard DCGAN (Radford et al., 2016), for example, consists of basic convolutional layers with batch norm and ReLU transformations. By contrast, Real NVP requires a scheme of coupling, masking, reshaping, and factoring over variables. Our proposed methods can be applied to any GAN, so that they can leverage any improvements made in new GAN architectures.

Extracting density estimates from GANs presents several challenges. A (non-invertible) GAN learns an embedding of a lower-dimensional latent space (the random codes) into a much higher dimensional space (the space of all possible images of a certain size).

Thus, the probability distribution that it learns is restricted to a low-dimensional manifold within the higher-dimensional space. Exact densities for images can be computed via the Jacobian if the latent code is known, as we will show in the next section, but densities are technically zero for images that are not exactly generated by any latent code. Extending densities meaningfully beyond the data manifold requires either incorporating an explicit noise model, such as in the recent Entropic GAN (Balaji et al., 2018), or learning a projection from images to latent codes, such as in BiGAN (Donahue et al., 2016).

We avoid these complexities by creating a simple regressor network that accepts an image and returns its estimated probability density. Training such a regressor network is easy if one has a large dataset of images labeled with their probability densities. In Section 2.2 we describe a simple method for obtaining such a dataset.

2.2 Extracting probability densities

A GAN generator *G* takes a random variable *Z* with a known latent distribution P_Z and produces an image *G(Z)* from an implicit learned distribution P_d. But what is P_d? If *G* is differentiable and bijective, then for *x=G(z)* the change of variables formula (Munkres, 2018) yields the exact density of the warped distribution. For *x = G(z)* we have

$$P_d(x) = P_z(G^{-1}(x))|\det \partial G^{-1}(x)|,$$

where

$$\partial G^{-1}(x)$$

is the Jacobian of the inverse function at *x*.

But most GAN generators are not bijective; they map a low-dimensional latent space to a high-dimensional pixel space, so the Jacobian is not square and we cannot compute a determinant. The solution is to perform calculations not on the codomain, but on the low-dimensional manifold consisting of the image of the latent space under *G*. If *G* is differentiable and injective, then this manifold has the same intrinsic dimensionality as the latent space, and we can consider how a unit cube in the *n*-dimensional latent space distorts as it maps onto the (also *n*-dimensional) image manifold. The resulting modified formula is

$$P_d(x) = P_z(z)|\det \partial G^{\mathrm{T}}(z)\partial G(z)|^{-\frac{1}{2}}. \quad (2)$$

The formula uses the fact that $\det(M^{-1}) = (\det M)^{-1}$ for any square matrix *M*. It also uses the fact that the squared volume of a parallelpiped in a linear subspace is computed by projecting to subspace coordinates via the transpose of the coordinate matrix, resulting in the square matrix

$$\partial G^{\mathrm{T}}\partial G$$

(an expression which is known as a *metric tensor*), and then taking the determinant.

The Jacobian can be computed analytically from the network computation graph, or numerically via a finite difference approximation (we found that the latter approach was much faster and did not change the qualitative results). Once computed, we can find the above determinant via a QR decomposition. If

$$\partial G = Q \cdot R$$

where *Q* is an *mxn* matrix with orthonormal columns and *R* is an *nxn* upper-triangular matrix, then

$$\det \partial G^{\mathrm{T}} \partial G = \det R^{\mathrm{T}} Q^{\mathrm{T}} Q R = \det(R)^2 = \left(\prod_{i=1}^{n} r_{ii} \right)^2 \quad (3)$$

Substituting back into equation (2), we obtain the probability formula

$$P_d(x) = P_z(z) \prod_{i=1}^{n} |r_{ii}|^{-1} \quad (4)$$

In practice, we use the log-densities to avoid numerical over/underflow.

To generalize probability predictions to novel images, we train a separate regressor network on samples from *G*, which are labeled with their log-probability densities. This regressor predicts densities directly from images. We will refer to this as the pixel regressor. This regressor does not truly learn a probability distribution, but is a reasonably accurate proxy.

Our basic generative model was a DCGAN (Radford et al., 2016), and the structure of our pixel regressor was modified from the discriminator.

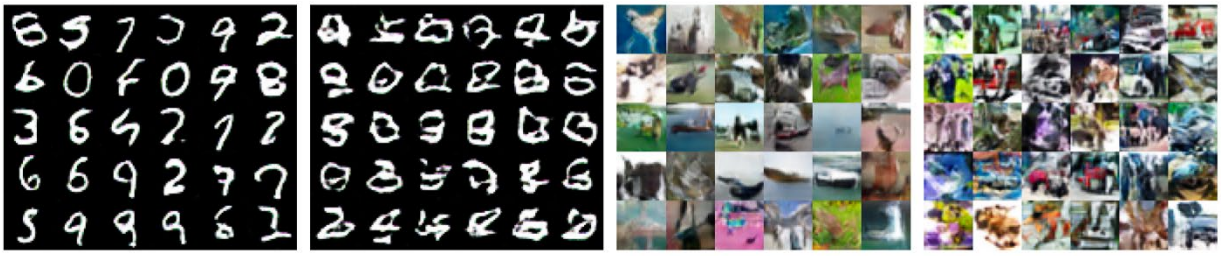

Figure 2 From left to right: random samples from a GAN trained on MNIST, samples of lowest probability density according to the pixel regressor, random samples from a GAN trained on CIFAR, samples of lowest density according to the pixel regressor.

2.3 Sanity check: do GANs yield reasonable probability estimates?

The accuracy of GAN-based density estimation depends on the accuracy of the generated density labels, and the ability of the regression network to generalize to unseen data. In this section, we investigate whether the obtained probability densities are meaningful. We do this quantitatively by comparing histograms of predicted densities in the train and test datasets, and also qualitatively by examining how probability density correlates with image quality.

2.4 Comparing Histograms

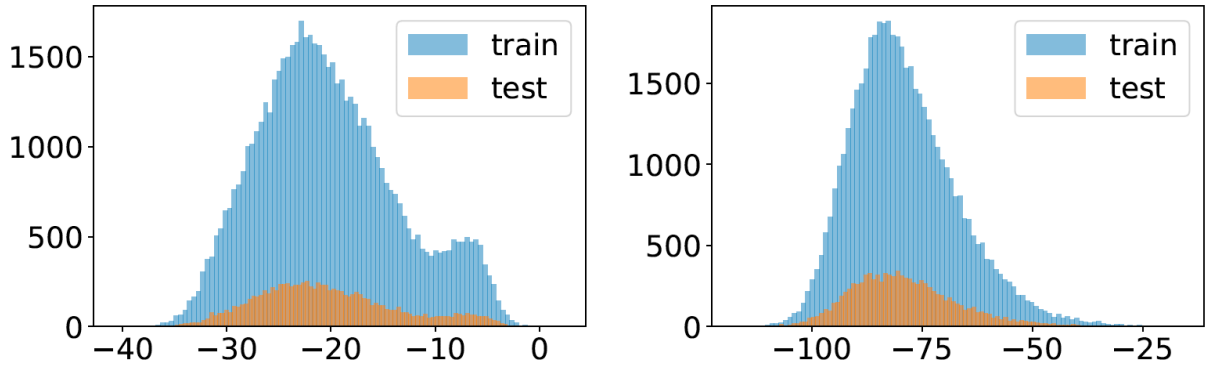

Figure 3 **Left:** histogram of log probability densities of MNIST train and test data as predicted by a pixel regressor for an MNIST GAN. **Right:** histogram of log densities of CIFAR train and test data as predicted by a pixel regressor for a CIFAR GAN.

The GAN and regressor model can be inaccurate because of under-fitting (e.g., missing modes), or overfitting (assigning excessively high density to individual images). We test for these problems by plotting histograms for the probability densities on both the train and test data to validate that these distributions have high levels of similarity.

Results are shown in Figure 3. The test histograms appear as a scaled-down version of the train histograms because the test sets contain fewer samples (we did not normalize the histograms by number of samples because this difference in scale helps in seeing both distributions on the same figure). For both MNIST and CIFAR, we see a very high degree of similarity between test and train distributions, indicating a good model fit (without over-fitting).

2.3.1 Visualizing typical and low density images

We get a stronger sense for what the density estimator is doing by visualizing ``outliers'' that have low probability density. Figure 2 shows typical samples produced by the GAN models for MNIST and CIFAR. We see that the GANs fit the distributions nicely, as typical samples reflect what we want these images to look like. However, the lowest density outliers (selected from 50,000 GAN random samples) are extremely irregular and clearly lie away from the modes of the distribution. When we use more sophisticated GANs, such as StackGAN or the recent StyleGAN (Figure 1) the low

density images always contain more complex textures and varied features, while the high density images are very uniform (as we will discuss further in the next section).

These visualizations suggest that GAN-based density estimators make reasonable density predictions. However, we will see in the next section that even highly accurate density estimation can have unreasonable consequences for some tasks.

2.4 Be careful what you wish for: the difficulties of interpreting image densities

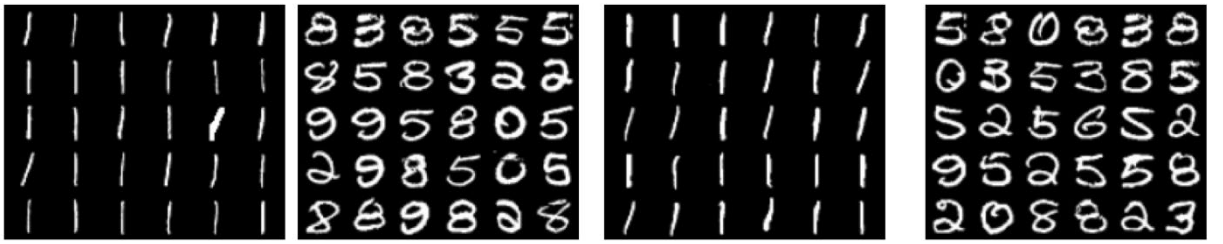

Figure 4 Highest and lowest density real MNIST digits as predicted by a pixel regressor for a GAN trained on MNIST with and without 1s and tested on MNIST with 1s.

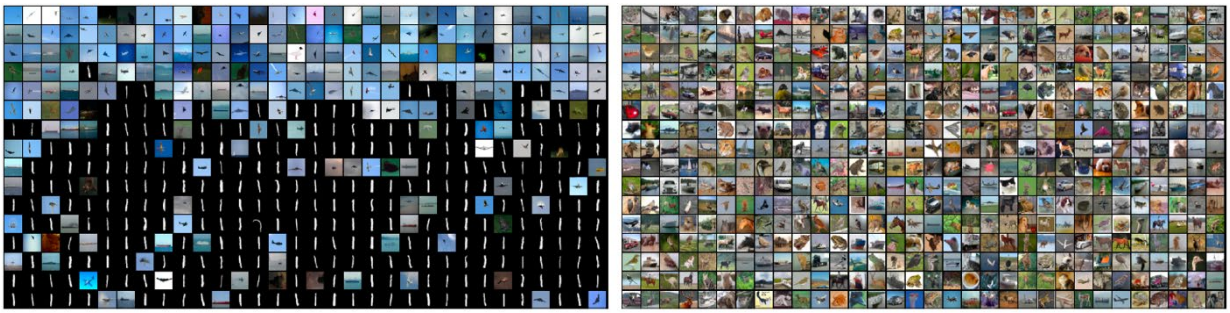

Figure 5 **Left:** Highest density 512 images from CIFAR and MNIST combined, as predicted by a pixel regressor trained on CIFAR. **Right:** Highest density 512 images for the combined data as predicted by a code regressor for a GAN trained on CIFAR.

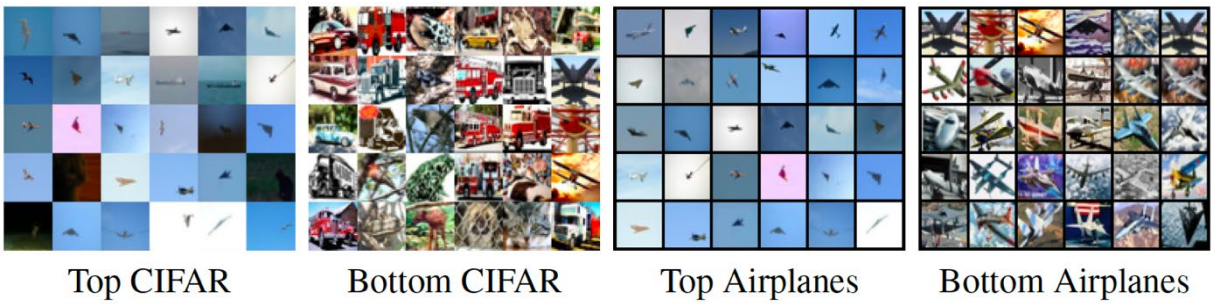

Figure 6 From left to right: Highest density real CIFAR images according to a pixel regressor trained on CIFAR, lowest density CIFAR images for the same distribution, highest density real airplanes, lowest density airplanes.

Although in some respects the learned probability densities correlate sensibly with image complexity and quality, we now show that these distributions are also highly irregular and non-uniform -- a characteristic that makes them difficult to interpret. In particular, the densities do not correlate well with human intuitions about semantic categories, such as object class or digit type.

2.4.1 Everybody loves 1s

We saw in Section 2.3.1 that image densities could be used to discern certain kinds of visual outliers from an image distribution. But what about the inliers? In this section we dive further into what image characteristics most strongly determine image density.

The left two images of Figure 4 show the most likely and least likely real images from the MNIST dataset. We see that all of the most likely images are 1s, while all of the least likely images are more ``loopy'' digits. This may seem unintuitive at first: 1's are just as likely to occur as any other digits, so why should they have higher probability density? However, this preference for 1s is in fact the result of *correct* density estimation; images of 1s in the MNIST dataset are all very well aligned and similar, and when interpreted as vectors in a high-dimensional space they cluster closely together. As a result, the 1s define an extremely high-density mode in the image distribution. This can be seen prominently in Figure 3, where we found that the bump on the right of the MNIST histogram was almost exclusively comprised of 1s.

To better assess the severity of this problem, we train the density estimator on all images except 1s. Intuitively, the 1s should now be outliers from the distribution. However, the density function still thinks the opposite. When the density of this incomplete distribution is evaluated on all MNIST test data (including the 1s), the most likely digits are still 1s (Figure 4, second image from the right).

This effect is likely because of the constant black background in images of 1s. Most pixels in these images are black (the most common value), and so these images lie relatively close (in the Euclidean sense) to many other MNIST images, making them inliers rather than outliers.

A similar problem manifests in the CIFAR dataset (Figure 6). In this case, the most likely images contain a simple blue background. This is likely because the ``airplane'' class contains many images with a smooth background of a similar blue color, and so these images lie close together in Euclidean distance, defining a high-density mode. Furthermore, in images of high density, the actual object of interest is extremely small and the background is dominant. Images with large foreground objects or complex backgrounds contain high-frequency features that do not correlate as well, so they lie far apart in Euclidean space and have relatively low densities as in Figure 6.

2.4.2 Are CIFAR images outliers in their own distribution?} \label{outliers}

Intuitively, one might expect to use density estimates for outlier detection; outliers from other, highly distinct distributions should have extremely low densities compared to inliers. We saw in Section 2.3.2 that densities were able to detect irregular/outlier images sampled from the learned distribution.

We study the task of deciding whether MNIST images are outliers from the CIFAR distribution. To this end, we train a density model on only CIFAR, and evaluate the density function on both CIFAR and MNIST images. The most likely images from the combined CIFAR/MNIST dataset are shown in the left image of Figure 5. We see that the set of most likely images is dominated by MNIST digits, with a small number of extremely simple CIFAR images in the top as well. Histograms of these densities are depicted in the leftmost image of Figure 8, and we see that MNIST is indeed far more likely than CIFAR.

This result is consistent with the experiments above -- smooth, geometrically structured images lie far to the right of the distribution. The MNIST images apparently lie in an extremely high density mode. However, in the CIFAR distribution, highly structured images of this type seldom appear. This indicates that the high density region occupies an extremely small volume and thus very small probability mass. Meanwhile, the lower-density outlying region (which contains the vast majority of the CIFAR images) comprises nearly all the probability mass.

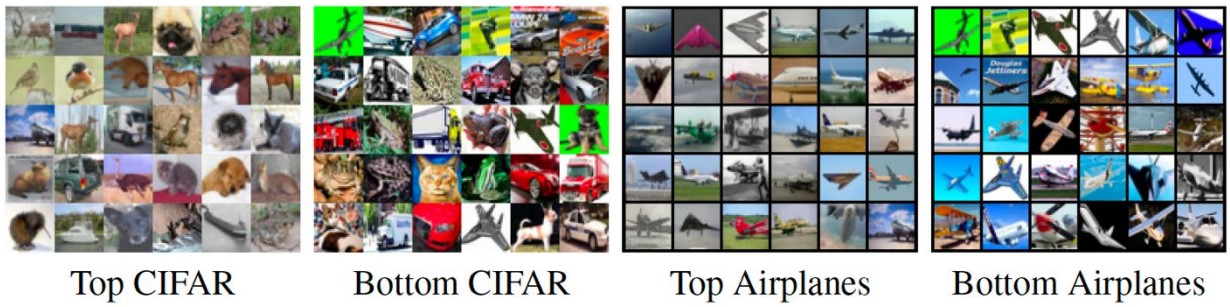

Figure 7 Highest and lowest density real CIFAR images, as predicted by a latent code regressor trained on CIFAR, and highest and lowest density CIFAR airplanes as predicted by a latent code regressor.

2.5 Making density functions interpretable

The experiments in Section 2.4 indicate that probability densities on complex image datasets have a structure that corresponds more to certain geometric properties of images than human-recognizable categories. Fairly ``typical" images often lie far from the modes of the distribution, their probability mass spread thinly. This lack of interpretability is a consequence of a well-known problem; the Euclidean distance between images does not capture an intuitive or semantically meaningful concept of similarity. ``Outliers" of a distribution are points that lie far from the modes in a *Euclidean* sense, and so we should expect the relationship between density and semantic structure to be tenuous.

To make density estimates interpretable, we need to embed images into a space where Euclidean distance has semantic meaning. We do this by embedding images into a deep feature space. In deep feature space, nearby images have similar semantic structure, and well separated images are semantically different. This enables distributions to have interpretable modes and outliers.

There are many options to choose from when selecting a deep embedding.
In the unsupervised setting where we already have a GAN at our disposal, the simplest choice for a feature embedding is to associate images with their latent representation *z,* the pre-image of the GAN. This embedding is particularly simple because the density function in this space is simply the Gaussian density, which can be evaluated in closed form. We learn this density mapping by associated each image with the density of its pre-image *z,* without accounting for the Jacobian of the mapping (using a GAN trained with a slightly modified, InfoGAN-based loss (Chen et al., 2016) as described in the supplementary material, to impose more structure on the latent space). For this approach, we train a regressor to predict the density of the z code that generated an image; we refer to this as the ``code regressor."

2.5.1 Images are now inliers in their own distribution
We show the most and least likely CIFAR images under the deep feature model in Figure 7. Unlike the pixel-space model, there is now diversity in the most likely images, and the distribution is not dominated by blue sky. The deep model also produces much more uniform densities than the pixel model, as shown in the rightmost plot of Figure 8, where 1s no longer completely dominate the far right -- this is expected since the MNIST dataset is itself fairly uniform with few semantic outliers.

We saw in Section 2.4.2 that MNIST digits were inliers with respect to the CIFAR distribution, and many CIFAR images were outliers in their own distribution (when estimating densities in the pixel space). When we perform density estimation in deep feature space, density estimates capture a more intuitive notion of outliers. To show this, we train a deep feature density estimator on the CIFAR distribution only, and then infer densities on the combined CIFAR and MNIST dataset. The middle plot of Figure 8 shows the histogram of estimated densities. We see that CIFAR images now occupy high density regions close to the distribution modes, and MNIST images occupy the low density ``outlier" regions.

The right image in Figure 5 shows the most probable CIFAR and MNIST images with respect to the CIFAR distribution. Unlike the left image in the same figure, we now see that all of the most likely images are from the CIFAR distribution.

2.5.2 Deep densities depend on image content rather than smoothness

Unlike the pixel-space density estimator depicted in Figure 6, the deep feature model (Figure 7) favors images where the foreground object is well-defined and occupies a large fraction of the image. The least likely images contain many objects in unusual configurations or strange backgrounds (e.g., airplanes with a green sky).

For the category of airplanes shown in the two rightmost images of Figure 7 we see that the densities seem to no longer depend strongly on the image complexity, but rather on the image content and coloration.

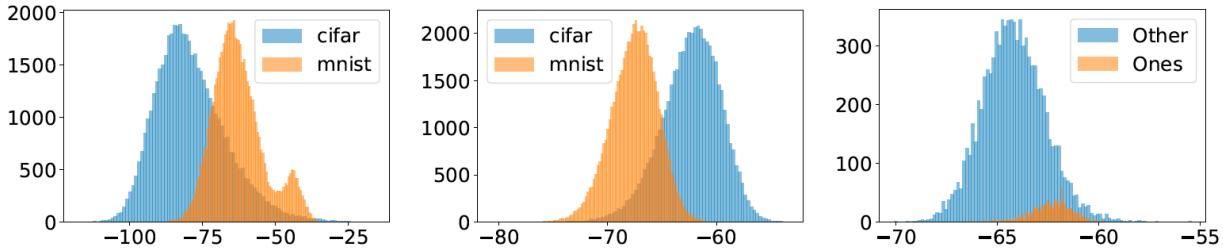

Figure 8 ***Left:*** *histogram of log probability densities of MNIST and CIFAR, predicted using a pixel-space density estimator for CIFAR.* ***Middle:*** *histogram of log densities of MNIST and CIFAR, predicted using the latent code regressor for a GAN trained on CIFAR.* ***Right:*** *histogram of log densities of MNIST, as predicted by a latent code regressor for a GAN trained on MNIST. Note that the log density values are much more clustered than in pixel space, though they are still near the top of the distribution.*

2.6 Conclusion

Using the power of GANs, we explored the density functions of complex image distributions. Unfortunately, inliers and outliers of these density functions cannot be readily interpreted as typical and atypical images, at least according to human intuition. However, we suggest that this lack of interpretability could be mitigated by considering the probability densities not of the images themselves, but of the latent codes that produced them. We postulate that such feature embeddings tend to cluster images in space around more semantically meaningful categories, consolidating probability mass that would otherwise be spread out thinly in pixel space due to many visual variations of the same type of object. There are a host of potential applications for the resulting image PDFs, including detecting outliers and domain shift that will be explored in future work.

3. Stabilizing Adversarial Nets with Prediction Methods

To support our research on using GANs to represent probability distributions of images, we have also been developing novel methods for training GANs and other adversarial nets more stably. Adversarial networks play an important role in a variety of applications, including image generation (Zhang et al., 2017; Wang & Gupta, 2016), style transfer (Brock et al., 2017; Taigman et al., 2017; Wang & Gupta, 2016; Isola et al., 2017) domain adaptation (Taigman et al., 2017; Tzeng et al., 2017; Ganin & Lempitsky, 2015), imitation learning (Ho et al., 2016), privacy (Edwards & Storkey, 2016; Abadi and Andersen, 2016), fair representation (Mathieu et al., 2016; Edwards and Storkey, 2016) etc. One particularly motivating application of adversarial nets is their ability to form generative models, as opposed to the classical discriminative models (Goodfellow et al., 2014; Radford et al., 2016; Denton et al., 2015; Mirza &

Osindero, 2014). We use such generative models to represent probability distributions of images, allowing us to adaptively use visual data to respond to changing circumstances.

While adversarial networks have the power to attack a wide range of previously unsolved problems, they suffer from a major flaw: they are difficult to train. This is because adversarial nets try to accomplish two objectives simultaneously; weights are adjusted to maximize performance on one task while minimizing performance on another. Mathematically, this corresponds to finding a *saddle point* of a loss function - a point that is minimal with respect to one set of weights, and maximal with respect to another.

Conventional neural networks are trained by marching down a loss function until a minimizer is reached (Figure 9a). In contrast, adversarial training methods search for saddle points rather than a minimizer, which introduces the possibility that the training path ``slides off'' the objective functions and the loss goes to negative infinity. (Figure 9b), resulting in ``collapse'' of the adversarial network. As a result, many authors suggest using early stopping, gradients/weight clipping (Arjovsky et al., 2017), or specialized objective functions (Goodfellow et al., 2014; Zhao et al., 2017; Arjovsky et al., 2017) to maintain stability.

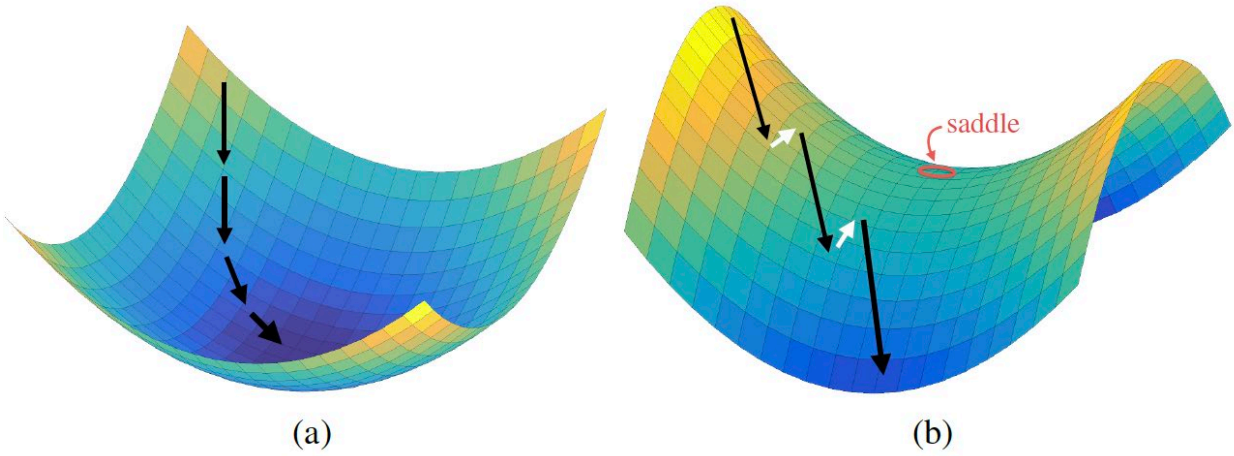

Figure 9 A schematic depiction of gradient methods. (a) Classical networks are trained by marching down the loss function until a minimizer is reached. Because classical loss functions are bounded from below, the solution path gets stopped when a minimizer is reached, and the gradient method remains stable. (b) Adversarial net loss functions may be unbounded from below, and training alternates between minimization and maximization steps. If minimization (or, conversely, maximization) is more powerful, the solution path ``slides off'' the loss surface and the algorithm becomes unstable, resulting in a sudden ``collapse'' of the network.

In our work, we present a simple ``prediction'' step that is easily added to many training algorithms for adversarial nets. We present theoretical analysis showing that the proposed prediction method is asymptotically stable for a class of saddle point problems. Finally, we use a wide range of experiments to show that prediction enables

faster training of adversarial networks using large learning rates without the instability problems that plague conventional training schemes.

Saddle-point optimization problems have the general form

$$\min_u \max_v \mathcal{L}(u, v)$$

for some loss function $\mathcal{L}$ and variables *u* and *v*. Most authors use the alternating stochastic gradient method to solve saddle-point problems involving neural networks. This method alternates between updating *u* with a stochastic gradient *descent* step, and then updating *v* with a stochastic gradient *ascent* step. When simple/classical SGD updates are used, the steps of this method can be written

$$\begin{aligned} u^{k+1} &= u^k - \alpha_k \mathcal{L}'_u(u^k, v^k) \quad | \quad \text{gradient descent in } u, \text{ starting at } (u^k, v^k) \\ v^{k+1} &= v^k + \beta_k \mathcal{L}'_v(u^{k+1}, v^k) \quad | \quad \text{gradient ascent in } v, \text{ starting at } (u^{k+1}, v^k) \,. \end{aligned} \tag{6}$$

Here, α_k and β_k are learning rate schedules for the minimization and maximization steps, respectively. The vectors $\mathcal{L}'_u(u,v)$ and $\mathcal{L}'_v(u,v)$ denote (possibly stochastic) gradients of $\mathcal{L}$ with respect to *u* and *v*. In practice, the gradient updates are often performed by an automated solver, such as the Adam optimizer (Kingma and Ba, 2015), and include momentum updates.

We propose to stabilize the training of adversarial networks by adding a *prediction* step. Rather than calculating v^{k+1} using u^{k+1} we first make a prediction, $\bar{u}^{k+1}$ about where the *u* iterates will be in the future and use this predicted value to obtain v^{k+1}.

Prediction Method

$$\begin{aligned} u^{k+1} &= u^k - \alpha_k \mathcal{L}'_u(u^k, v^k) \quad | \quad \text{gradient descent in } u, \text{ starting at } (u^k, v^k) \\ \bar{u}^{k+1} &= u^{k+1} + (u^{k+1} - u^k) \quad | \quad \textit{predict} \text{ future value of } u \\ v^{k+1} &= v^k + \beta_k \mathcal{L}'_v(\bar{u}^{k+1}, v^k) \quad | \quad \text{gradient ascent in } v, \text{ starting at } (\bar{u}^{k+1}, v^k) \,. \end{aligned} \tag{7}$$

The Prediction step (7) tries to estimate where *u* is going to be in the future by assuming its trajectory remains the same as in the current iteration.

We have performed a theoretical analysis and a wide range of experiments to demonstrate the benefits of the proposed prediction step for adversarial nets. We consider a saddle point problem on a toy dataset constructed using MNIST images, and

then move on to consider state-of-the-art models for three tasks: GANs, domain adaptation, and learning of fair classifiers. Further details of this work can be found in Yadav et al. (2017).

4. Inferring Granger Causality with Weak Supervision

Understanding causality lies at the heart of intelligent behavior. Knowing what causes an outcome allows us to understand how to modify or preserve elements of our behavior to encourage or avoid that outcome. Understanding the cause of our beliefs allows us to modify these beliefs when circumstances change. Causal knowledge allows us to better cope with changing circumstances, as it allows us to tell whether changes in an environment should affect the conclusions we can draw from past experience. While there has been considerable research on the theory of causality (Pearl, 2009) and the application of causal reasoning in a variety of domains (Guo et al., 2018; Yao et al., 2020), deriving a causal understanding of visual events has still been little explored.

4.1 Introduction

We have been studying this problem by focusing on the question of what components of a video indicate events causally related to a specific outcome. For example, suppose a video of a car driving ends in an accident. We will ask: when did the cause of the accident occur? (Of course, as real-world events have many causes, the exact answer will depend on framing and context, but highly salient events can be identified at points where predictive models suddenly change their predictions). We will also determine what parts of the images show a potential cause of the accident. So, when an accident is caused by another car running a red light, we may determine that the accident may have been caused {by factors captured within} a spatial and temporal piece of the video containing the car running the light. Furthermore, our framework will allow us to determine the cause of risks, even if they don't end in a particular outcome, such as when something happened that might have caused an accident but didn't. This work will take an important first step in building a causal understanding of videos. It has immediate applications in the forensic analysis of events, and in training. For example, these methods could be used to determine when a pilot in a flight simulator took an action that might have led to a crash.

We approach this problem using machine learning (ML). An ML approach faces the problem of how to gather appropriate training data to allow for the inference of causality. The notion of causality that we investigate first is that of Granger causality (Granger 1969). A Granger cause of some event B is a prior event A that is predictive of B; that is, if A is present, we tend to predict B, and if A is not present, we tend not to predict B. This is a weakened notion of causality as a time-directed correlation. As the old phrase goes, ``correlation does not equal causation'', but in the case of time series analysis it is often good enough. Furthermore, identifying a Granger cause can be the first step towards identifying a true underlying cause, in the full counterfactual sense. The main

insight behind our approach is that Granger causal information can be learned from observational video data that is only very weakly supervised, using insights from reinforcement learning.

We will explain this approach in the simplest case in which there are no actions (or, depending on perspective, one trivial action of moving forward in time), and a video is simply modeled as a Markov process. Furthermore, we will consider the case in which the reward is just a binary value given at the end of the video, depending on whether a particular outcome has occurred (eg., crash or no crash). These ideas can be easily extended to handle multiple possible outcomes, and reward that might occur at multiple points in the video.

In this setting, we borrow from reinforcement learning the concept of a value function (eg., Sutton and Barto, 2018). Given a state, the value function tells us the expected discounted reward. In our simple setting, the value function simply tells us the probability that the video will end with a particular outcome, perhaps discounted by the amount of time it will take for the outcome to be determined. This is equivalent to the notion of win probability used in sports. For example, ESPN provides real time win probability during a baseball game, which indicates the probability that either team will win, given the current state of the game.

Reinforcement learning provides many algorithms for learning a value function from sequential data in which only the reward is known (Sutton and Barto, 2018). In our case, this means that a video only needs to be labeled with a binary label indicating the outcome. However, once we have learned the value function, we can use it to provide fine-grained information about when important causal events have occurred. For example, suppose we have a video of a car driving. At first, the probability of an accident is close to zero. But at some point, the probability shoots up to .8. This tells us that something has occurred that is likely to cause an accident. Inference about the value function will be done with a neural network. So we can also use visualization methods developed for neural networks to determine what portion of the image caused this increase in probability (eg., another car running a red light). Note that this information not only tells us about the cause of accidents, it also tells us when something has occurred that might have caused an accident, even if it didn't ultimately occur.

4.2 Pilot Study

To make this approach more concrete, we will describe an initial implementation along with some preliminary results, using data from the driving game BeamNG. We have created a large set of videos captured from this game, some of which end in accidents, and some of which don't. From this video we aim to learn a value function:

$$v(s) = E\left(\sum_{k=0}^{\inf} \gamma^k R_{t+k+1} | s_t = s\right)$$

Here *s* denotes the state of the game. We characterize the state by concatenating the ten previous frames of video. Like much reinforcement learning, we assume a Markov process, so that the value is independent of any previous frames. *v(s)* denotes the value function. s_t denotes the state at time *t*. R_{t+k+1} denotes the reward received at time *t+k+1*. In our case, the reward will be given only at the end of the video and will be 1 if an accident has occurred and 0 if it hasn't. γ is a discount factor, which in our experiments is slightly less than 1. *E* indicates that we are computing the expected value of a state.

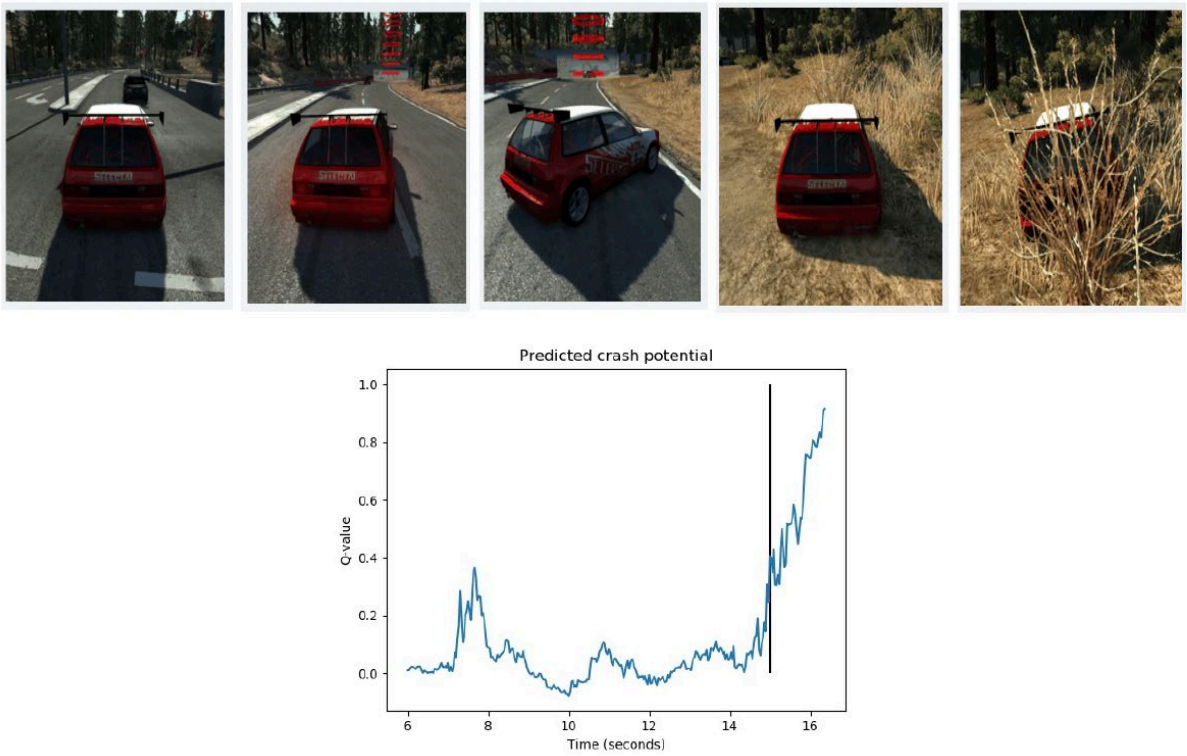

Figure 10 Above we show a sequence of images from a BeamNG video, in which a car swerves, causing it to crash into the woods. On the right we show probability of a crash occurring, as calculated by our proposed algorithm, for a similar video sequence. The vertical line shows a ground truth human judgement about when the cause of the crash occurred. We can see that at that point, the algorithm begins to predict a crash.

We adapt Q-learning to our setting (actually, since we do not have actions in this model, the distinction between several different learning approaches, such as on-policy and off-policy learning, disappears). We can learn the value function through the update:

$$v(s) \leftarrow v(s) + \alpha[(R_{t+1} + \gamma v(s_{t+1})) - v(s_t)]$$

where α is a learning rate. We can then adapt this to train a neural network (Mnih et al., 2013, 2015) as:

$$w_{t+1} = w_t + \alpha[(R_{t+1} + \gamma\hat{v}(s_{t+1})) - v(s_t))]\nabla\hat{v}$$

Here *w* represents the weights of a neural network that approximates the value function, and $\hat{v}(s)$ represents the output of this network for state *s*. Using this approach, we have successfully trained a neural network to determine the time at which the cause of an accident occurred. We show an example in Figure 10.

4.3 Future Work

We plan to expand this pilot work in a number of directions. First, we will examine the spatial dimension of the problem, determining where the cause occurred, and using this information to understand what actions are potential causes of events. Second, we will experiment with a variety of domains, using real-life video. Third, we will explore the use of these techniques in performing weakly supervised learning, which can be used to develop powerful features for video classification and prediction, based on only minimal supervision. Fourth, our current approach only identifies candidates for causality, using the notion of Granger causality (Granger, 1969). We will explore methods of teasing apart causality and correlation, through real or simulated interventions.

The value function allows us to identify when something occurred that significantly altered the likely outcome. To determine where that occurred, we will leverage sensitivity analysis methods used to visualize the saliency of different image components and their relevance to a final classification decision (Simonyan, 2013; Springenberg, 2014). These methods compute the gradient of the network output with respect to the input to determine which parts of the input are most relevant to the output. This will allow us to locate the most salient regions of the images. We can, analyze these regions by, for example, applying off-the-shelf object detectors (eg., Redmon & Farhadi, 2017) to determine what objects may be related to the outcome, or even use captioning systems (eg., Yu et al., 2016; Gao et al., 2017) to build verbal descriptions of possible causes.

BeamNG allows us to create fairly realistic video in which we control the course of events. However, we feel that it is also important to experiment with real-world video. One option is to apply our methods to highlight detection in sports video. For example, we can collect video of portions of soccer games that either do or do not end in a goal. If we can learn the value function for these games, we should learn that when the ball is centered near the goal, a goal becomes more likely. This can be used to find highlights in which the likelihood of a goal became high, regardless of whether or not a goal was actually scored.

We also hypothesize that by learning the value function, we will learn deep features that will be helpful in a range of other video analysis tasks. For example, suppose in our videos, car crashes are often caused by cars running red lights. To learn this, a neural network that learns the value function must develop features that implicitly allow it to detect cars and traffic lights. We therefore expect that in training a network based just on very weak supervision (eg., whether the video ends in an accident) we will learn

features that can be used to detect objects or brief events in video. If true, this has the potential to allow us to learn important new features using massive volumes of video data.

Finally, we note that our initial approach truly only learns Granger causality. This tells us whether some components of a time series can be used to predict subsequent events in the series. These features may, however, only be correlated with future outcomes, without being causally related. We see Granger causality as providing candidates for true causes. But to separate events that are correlated with a future outcome from those that cause it, we need the ability to perform interventions. With an intervention, we hold everything fixed, changing only one factor, to see if its variation produces a change in the outcome. This is done, for example, in blinded clinical trials, in which two statistically identical pools of patients are given either a new drug or a placebo. For simulated environments like BeamNG, we can control driving behavior, and determine which actions are causally related to an outcome. This would have applications in training systems, in which we want to ask, for example, what actions of a pilot in a flight simulator might have increased the probability of an accident. We will also explore methods for simulating interventions in real environments. For example, in a real driving video, by slowing the video we simulate what would have happened if the car was being driven more slowly. We can then use a learned value function to determine whether a slower car would have been less likely to have an accident.

4.4 Relation to Prediction

Our approach to computing Granger causality is closely related to the problem of predicting future events in video. By learning a value function, we learn the probability that a particular event will occur in the future. A primary difference is that we aim to learn to determine at what point in time a particular event becomes predictable. This creates a significant problem in identifying appropriate training data, which we solve using tools from reinforcement learning.

For example, Vondrick et al., (2016) predicts visual representations that will occur in a video a fixed δ of time in the future. This allows them to use unlabeled video for self-supervision. In our setting, however, even knowing when an event has occurred, we do not know when its cause occurred, so we do not know when it became predictable. We therefore don't know, for example, whether video two seconds before an accident is a positive training example, because the cause of the accident is apparent, or a negative example, because the cause has not yet happened. Other work on prediction avoids this problem by either treating points in time prior to the event has having some weighted connection to the event, with an exponential decay in the weight (Ryoo, 2011), so that more distant times are assumed less relevant to the event or apply a kind of curriculum learning to gradually increase the ability of a system to predict further in the future (see, eg., Suzuki et al., 2018). In principle our approach should be more flexible in learning to determine relations between events that are arbitrarily far apart in time, although these prior methods will provide good points of comparison.

5. Semantic Segmentation and Domain Shift

We have also explored a concrete example of solving computer vision problems in the face of domain shift. In this work, we explore the problem of semantic segmentation, in which each pixel in an image is labeled according to the type of object it comes from (eg., sidewalk, road, car). We have developed a framework for semantic segmentation that dynamically adapts to changing environments over time. This work also handles the problem of forgetting knowledge from past environment. We introduce a memory that stores feature statistics from previously seen domains. These statistics can be used to replay images in any of the previously observed domains, thus preventing catastrophic forgetting. This work has been published in ICCV 2019 (Wu et al., 2019), and can be accessed at:

https://openaccess.thecvf.com/content_ICCV_2019/papers/Wu_ACE_Adapting_to_Changing_Environments_for_Semantic_Segmentation_ICCV_2019_paper.pdf

6. Other work

In addition to the work described in this report, our L2M funding has supported us in a number of other pieces of work. Below we provide a list of publications supported in part by L2M.

- Ace: Adapting to changing environments for semantic segmentation. Wu, Zuxuan, Xin Wang, Joseph E. Gonzalez, Tom Goldstein, and Larry S. Davis. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2121-2130.

- Understanding the (un) interpretability of natural image distributions using generative models. Krusinga, Ryen, Sohil Shah, Matthias Zwicker, Tom Goldstein, and David Jacobs. arXiv preprint arXiv:1901.01499 (2019).

- Understanding generalization through visualizations. WR Huang, Z Emam, M Goldblum, L Fowl, J Terry, F Huang, T Goldstein arXiv preprint.

- Truth or backpropaganda? An empirical investigation of deep learning theory. _M Goldblum, J Geiping, A Schwarzschild, M Moeller, T Goldstein International Conference on Learning Representations (ICLR), 2020.

- Adversarially Robust Distillation. M Goldblum, L Fowl, S Feizi, T Goldstein_AAAI Conference on Artificial Intelligence.

- Visualizing the Loss Landscape of Neural Nets._H Li, Z Xu, G Taylor, T Goldstein_Neural Information Processing Systems (NIPS), 2018.

- Adversarially robust transfer learning. A Shafahi, P Saadatpanah, C Zhu, A Ghiasi, C Studer, D Jacobs, T Goldstein_International Conference on Learning Representations (ICLR), 2020.

- Adversarial Training for Free!_A Shafahi, M Najibi, A Ghiasi, Z Xu, J Dickerson, C Studer, L Davis, G Taylor, T Goldstein_Neural Information Processing Systems (NeurIPS).

- Transferable Clean-Label Poisoning Attacks on Deep Neural Nets. C Zhu, WR Huang, H Li, G Taylor, C Studer, T Goldstein_International Conference on Machine Learning (ICML), 2019.

- Are Adversarial Examples Inevitable?_Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, Tom Goldstein_International Conference on Learning Representations (ICLR), 2019.

- Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks_. A Shafahi, WR Huang, M Najibi, O Suciu, C Studer, T Dumitras, T Goldstein_Neural Information Processing Systems (NIPS), 2018.

- Linear Spectral Estimators and an Application to Phase Retrieval. R Ghods, A Lan, T Goldstein, C Studer_International Conference on Machine Learning (ICML), 2018.

7. References

Martín Abadi and David G Andersen. Learning to protect communications with adversarial neural cryptography. arXiv preprint arXiv:1610.06918, 2016.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In ICML, 2017.

Yogesh Balaji, Hamed Hassani, Rama Chellappa, and Soheil Feizi. Entropic gans meet vaes: A statistical approach to compute sample likelihoods in gans. arXiv preprint arXiv:1810.04147, 2018.

Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. IEEE, 2008.

Andrew Brock, Theodore Lim, JM Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. In ICLR, 2017.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in neural information processing systems, pp. 2172–2180, 2016.

C Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. IEEE transactions on Information Theory, 14(3):462–467, 1968.

Jeremy S De Bonet and Paul Viola. Texture recognition using a non-parametric multi-scale statistical model. In Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, pp. 641–647. IEEE, 1998.

Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In NIPS, 2015.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016.

Jeff Donahue, Philipp Kr.henbühl, and Trevor Darrell. Adversarial feature learning. In ICLR, 2016.

Harrison Edwards and Amos Storkey. Censoring representations with an adversary. In ICLR, 2016.

Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In iccv, pp. 1033. IEEE, 1999.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Proceedings of The 32nd International Conference on Machine Learning, pp. 1180–1189, 2015.

Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. Video captioning with attention-based lstm and semantic consistency. IEEE Transactions on Multimedia, 19(9): 2045–2055, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NIPS, pp. 2672–2680, 2014.

Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. Econometrica: Journal of the Econometric Society, pp. 424–438, 1969.

Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Bridging implicit and prescribed learning in generative models. arXiv preprint, 2017.

Ruocheng Guo, Lu Cheng, Jundong Li, P. Richard Hahn, and Huan Liu. A survey of learning causality with data: Problems and methods, 2018.

Nils-Bastian Heidenreich, Anja Schindler, and Stefan Sperlich. Bandwidth selection for kernel density estimation: a review of fully automatic selectors. AStA Advances in Statistical Analysis, 97(4):403–433, 2013.

Jonathan Ho, Jayesh Gupta, and Stefano Ermon. Model-free imitation learning with policy optimization. In International Conference on Machine Learning, pp. 2760–2769, 2016.
Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In CVPR, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948, 2018.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In NIPS, pp. 5041–5049, 2016.

Marwan A Mattar and Erik G Learned-Miller. Improved generative models for continuous image features through tree-structured non-parametric distributions. UMass Amherst Techinical Report 06-57, 2006.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

James R Munkres. *Analysis on manifolds*. CRC Press, 2018.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? arXiv preprint arXiv:1810.09136, 2018.

Bruno A Olshausen and David J Field. Natural image statistics and efficient coding. Network: computation in neural systems, 7(2):333–339, 1996.

George Papamakarios, Iain Murray, and Theo Pavlakou. Masked autoregressive flow for density estimation. In Advances in Neural Information Processing Systems, pp. 2338–2347, 2017.
Dong-Chul Park. Image classification using naïve bayes classifier. Int. J. Comp. Sci. Electron. Eng.(IJCSEE), 4, 2016.

Emanuel Parzen. On estimation of a probability density function and mode. The annals of mathematical statistics, 33(3):1065–1076, 1962.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, second edition, 2009.

Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Transactions on Image processing, 12(11):1338–1351, 2003.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016.

Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263–7271, 2017.

Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 2, pp. 860–867. IEEE, 2005.

Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In 2011 International Conference on Computer Vision, pp. 1036–1043. IEEE, 2011.

Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 693–700, 2010.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In International Conference on Machine Learning, pp. 1218–1226, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034, 2013.

Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806, 2014.
Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

Tomoyuki Suzuki, Hirokatsu Kataoka, Yoshimitsu Aoki, and Yutaka Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident db. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3521–3529, 2018.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In ICLR, 2017.

Radu Timofte, Tinne Tuytelaars, and Luc Van Gool. Naive bayes image classification: beyond nearest neighbors. In Asian Conference on Computer Vision, pp. 689–703. Springer, 2012.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In ICLR Workshop, 2017.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 98–106, 2016.

Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In ECCV, pp. 318–335, 2016.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Wu, Zuxuan, Xin Wang, Joseph E. Gonzalez, Tom Goldstein, and Larry S. Davis. "ACE: adapting to changing environments for semantic segmentation." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2121-2130. 2019.

Abhay Yadav, Sohil Shah, Zheng Xu, David Jacobs, and Tom Goldstein. Stabilizing adversarial nets with prediction methods. arXiv preprint arXiv:1705.07364, 2017.

Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. A survey on causal inference, 2020.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. CoRR, abs/1506.03365, 2015. URL http://arxiv.org/abs/1506.03365.

Haonan Yu, JiangWang, Zhiheng Huang, Yi Yang, andWei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4584–4593, 2016.
Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. arXiv preprint, 2017a.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In ICCV, 2017b.

Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. In ICLR, 2017.