[SQUEAKING]

[RUSTLING]

[CLICKING]

SARA BEERY: We're going to finish up our short series on transfer learning today. And the other logistical thing I did want to bring up is I know that project proposals are due this Friday. During our instructor meeting today, we put together a rubric for those project proposals just to try to give you guys a little bit more structured understanding of what we're going to be evaluating those on. So that's up on Piazza. Essentially, it's like out of 10, and there's five different things that are each worth two points.

And then the other thing I wanted to say is that this is actually, shockingly, already my last lecture of the semester. I don't know where the time goes, but it has been really awesome getting to lecture you this semester. And I will probably see you towards the end of the semester anyway, but yeah. Time flies.

Cool. So today, continuing that thread on thinking about how we can transfer knowledge specifically within the constraints of these deep learning models, or these deep learning systems, today, we're going to be specifically talking about how we can transfer knowledge about the inputs to the system.

And so we're going to spend some time thinking about how you can think about generative models as another way to capture information about data inputs. And then we're also going to talk a little bit about models that learn to learn, so this idea of meta-learning and how you can build models that are almost designed to be good at transfer learning, designed to be good at sharing information or learning new things.

Cool. So if we're thinking about knowledge about the inputs, maybe one way to think about this is if you know something about the distribution of inputs you're going to expect, maybe there's a way to actually use that to direct or accelerate your learning or even directly target a specific type of learning.

So you have a data set, and then we spent quite a few lectures with Phil talking about this ability to learn how to generate data from that same distribution, this idea that you can use generative modeling. And so now, that gives you access to some model that then is able to sample from that distribution.

And so here, at a first glance, this might not seem that interesting. You had data, and then you used it to train a model that mimics the data that you already had. But you already had the data, so how does it actually help? How are you going to get more information out of a system that's been trained to model from a specific distribution?

But one of the interesting things that you can think about this as is, assuming that generative model is kind of very good, you can almost think about the model itself as a new way, or a new mechanism to access data, going beyond the capabilities of the initial training data set. So maybe you call it data plus plus. And where does that additive value that you're getting from a data generator that you might not get from the data itself?

And here's an interesting quote. So this is from the release of the first Stable Diffusion model. They said that, "This release is the culmination of many hours of collective effort to create a single file that compresses the visual information of humanity into a few gigabytes." And that sounds pretty awesome.

So essentially, it's the idea that, actually, a generative model is a really cool form of data compression. So you can take that generative model, and you can go from some latent variable to anything on Earth, like any sort of visual-- all visual information of humanity, then that is a shocking compression of everything. You think about even just the storage capacity of all the images on the internet versus the storage capacity of a Stable Diffusion model. That's a significant difference in terms of what you're capturing.

So here, how do we think about what data plus plus means, and how do we think about data as maybe like an object?

So if we start with some X, which is the set of the original training data, which is little x, z, which is a set of latent variables that can correspond to the original training data learned by our generative model, the mapping from z to x-- so this is basically the mapping that goes from some latent variable to a generated data point, and that's G. That's the generative model we've learned.

And then if you also have G inverse-- essentially, this inverse mapping that lets you go from data to its associated latent variable-- you can actually define these. You can take these and define operators over those objects that enable us to do things to the generator and the system of data plus generator of data that we couldn't easily do to the data itself.

So for example, you can think about interpolation over a data set. Pretty standard definition of interpolation. But now, you can think about maybe some-- if you have one set of this, like X, Mathcad X that's some data set, some latent space, a generator for that latent space to data in the inverse. If you have one of those, and you have another one that's for some of different data set, possibly, or a different generator, you can think about interpolating between the two, right?

So some interpolation between those two things would generate a new one. And you can think about manipulation, right? You can think about how you might add some bias term to that that would give you, again, some whole new kind of object for accessing and interacting with data.

You can think about composition. So here, how do you actually kind of compose two things is, of course, different from interpolating between two things. So now if you essentially are taking this space where you have just these different data products or data objects, you can think about really combining them and manipulating them in interesting ways.

You can also think about optimization, so how you might actually be able to explicitly find some set of generator data, latent space inverse generator, that's optimal for some given dimension of optimality. And you can imagine taking that and bringing it to bear on many, many different research domains or problem domains that people are interested in.

Things like graphics, visualization, data augmentation, counterfactual reasoning. And all of this work is kind of somewhat in progress, actually. So these are just some examples of papers that start getting at these mechanisms of interpolation, manipulation, composition, or optimization over generated data objects.

So if we think about this space of generative models-- so here, this is now just visualizing what we were just talking about. We have some set of latent variables. So these are maybe the controls, right? This is kind of the way that you control what you want to generate. And maybe you've made some reasonable constraints in your system so you can ensure that z is normally distributed.

Now, you have some generative model, G, that will take any point in that latent space and then synthesize an image. So here, now, if you sampled a different point in latent space, then you would sample a different image. And so now, given the relationship between dimensionality of movement in latent space and the image that's generated, you can think about this as a mechanism of control.

So maybe if you move along some controllable snippet of the manifold of natural images, you can explicitly do things like find some latent variable in your latent space, some dimension in that latent space that corresponds to something like pose for this bird or orientation in some way. And so maybe there's different dimensions of this control in latent space that corresponds specifically to disentangled factors of variation or change in the space of those real generated images. And so if you can find-- yeah?

AUDIENCE: On that slide, do you suppose that the two dimensions are linearly independent? And is it usually the case?

SARA BEERY: So your question is, are you supposing that these dimensions in this kind of latent variable space are linearly independent? So I think the assumption here is that there is some decomposable directions that are maybe some projection of this where you can find orthogonality, and that orthogonality then corresponds to some very clean specific variables of variation in the real image space.

In practice, sometimes, particularly for certain types or certain categories of generative models, this is true, but it's not necessarily easy to find. So people will do this cherry picked thing where they're like, hey, we found a dimension, where if we move in this dimension, you can see that it changes from day to night.

We'll talk more about some of these. But I think one of the challenges of this is, particularly as these latent spaces get higher dimensional, being able to really explicitly disentangle these dimensions of variation in a clean way can be very difficult and can be highly heuristic. Yeah.

Cool. So if you have that, if you have this kind of model that takes you from data to what we're calling data plus plus, and you can map now from your actual image now into the latent space itself-- so this is that G inverse-- now, you have the ability to ask these interesting counterfactual questions.

What would it look like if assuming that you have these measurable and actionable dimensions of variation? So here, now, you could say, OK. What would it look like if we moved around in this latent space? How do these different dimensions correspond to different maybe things within the manifold of real images?

So here's some work where you can actually show that you can improve your ability to categorize something by building an ensemble over different generated variations of that input data in some manifold. And so you get some, essentially, improved accuracy and robustness by taking a real input image and then figuring out where that corresponds to in latent space, and then moving around nearby in the neighborhood of that thing to find some maybe different poses, different orientations, slightly different realistic manipulations of that input image, and then taking an ensemble.

So it's almost like you're building some robustness into your categorizer based on some interesting dimensions of data augmentation, almost like test time augmentation within the generator itself, which is interesting. So there's some nice work that shows that, OK, actually, exploring dimensions of latent space can give you something that actually is more than the original training data.

Though I think an interesting and important point here is there's so much kind of knowledge baked into this. The fact that something nearby in that image space should be reasonably-- maybe the same category, not actually break a category boundary in some significant way, seems important. And I often try to point out, some of these assumptions that are built in, some of the ways they might fail.

So if you are just trying to understand something like cat, this is probably very reasonable. If you're trying to understand something about maybe a breed of cat, these dimensions of variation might start to get more confusing in terms of breaking that class boundary. Because to me, as someone who spends a lot of time thinking about animals, the first image and the one in the middle actually look like different breeds.

That difference in the morphology with that extra length of the ears? To me, if that's this type of cat, the fine grained level of categorization you're interested in, this might actually be more confusing and potentially break that robustness. So there's always some fundamental assumptions about what you're actually trying to do that are built in to the dimensions of variation that are good or bad for a given problem.

And then there's this question of, how do you discover these dimensions in a latent space, right? So one interesting thing you could think about is, if you're trying to discover an interesting dimension of variation in a latent space, there's ways you can do it experimentally. You take an image of something. You take an image of the same thing at night, for example, and then you look at where those both mapped to in latent space.

And then you decide, you calculate the vector of direction that goes from one to the other. So now, you've almost calculated a vector of direction that should correspond to this specific type of counterfactual. And so then you can think about explicitly finding things like dimensions of variation that zoom out on an image or zoom in on an image or brighten it or darken it.

So in this case, they were able to find corresponding specific shifts in the image that can be used to then control images and ones that are input agnostic. So now, you can find this optimal direction or dimension of variation by just kind of cropping images. And then you take an ensemble of the directions for each of those, and maybe you can find something experimentally.

And so here, you have these dimensions of zoom. You can find dimensions that correspond to shift even across different input sizes. You have dimensions that correspond to brightening an image. And then you have things like corresponding dimensions of darkening an image.

So here, this is interesting, because I would say that this is really capturing some very specific bias in our data. Because clearly, people are not taking pictures of volcanoes in the dark that aren't erupting. So when we try to make it dark, it thinks it needs to erupt, right? There's this clear bias that's just built into the training data.

And there are some other cool examples of this for other types of modalities as well. So there's actually a really famous example from word2vec, which is an embedding model for words like semantic meanings, where there's a direction in the latent space of models that will correspond to changing the tense of the word.

So you can move in that direction and go from swimming to swam and go from walking to walk by moving the same direction in that latent space for words. So these types of examples of disentangled dimensions of variation in latent space can be discovered for many modalities and have been discovered for many modalities.

And then there's kind of like these explicit latent space vectors that are disentangling these factors of variation. So you have things like winter to spring, or turning on the lights, going from day to night. The volcano eruption vector. Though, again, you'll note that you're more likely to get a more aggressive looking eruptions at nighttime than during the day.

So there's been a lot of research into how to efficiently discover these dimensions of variation within a latent space, these disentangled dimensions of variation, so let's look at that in a little bit more technical detail. So you can take a GAN, like a generative adversarial network, and you can walk in a straight line in a latent space and visualize what it looks like to walk in that straight line. And it kind of works, but it can be really cherry picking. So some of those directions correspond to stuff that just looks like trash or is just not easily disentangled into something that we can interpret.

And then there's some of fancier versions of a GAN, like StyleGAN, where they actually explicitly noticed that any layer of the model, any of those intermediate representations, also represents a latent space for the model. And so they determined that there's actually earlier layers in the model that are better, quote, unquote, latent representations than the z representation, that initial input representation.

And so then we'll try and talk about why that might be, why it might be better to actually think about the latent space from somewhere intermediate in the model versus explicitly with that input latent vector.

So if we have some natural image manifold, X, so some nonlinear data space, and now, we have some starting point, so this is the corresponding point in that data natural image manifold corresponding to this bird, this blue bird. Maybe you have another point here. This is corresponding to this fly.

So if you think about linear interpolation between these two points, arguably, you're moving off the manifold of real images. There's stuff in one end and in the other that are more realistic looking, but this direct linear interpolation in image space just gives you stuff in the middle that's basically these obvious kind of just additive images. And it doesn't look anything like any realistic natural world image would. It just doesn't match the statistics of real natural images.

So instead, if we have some sort of latent space that-- now, we've learned this data plus plus representation. So now, we have a latent space that's well-behaved, and we can map from that directly to that natural image manifold. Then, you can imagine that interpolating linearly in the latent space would correspond to a nonlinear interpolation in that data manifold space, but it would ensure that you're mapping within the manifold of real natural images the whole way. Because by definition, we've constructed something where any point in that latent space will map to the natural image manifold.

So now, if you do that same interpolation-- and this is from a paper called BigGAN back in 2018-- you do get stuff that at least looks a little bit more like natural images. But they might not really map to reality. You get some really weird things like this half bird, half fly thing where, basically, the model is just doing its best. It's doing its best to find some way to reasonably interpolate between these two things and stay within the constraints of that trained natural image manifold mapping.

But that z space, so this z space here where we're mapping from one point to another in a linear way, might not be the best way to organize the data to interpolate along. And so let's look at why that might be. So it goes back to the lecture from VAEs. If you guys remember, you have this data distribution that maps to a latent distribution, and you have a latent distribution that maps to a data distribution.

And so at the bottom, we're coloring the correspondence between those two spaces, essentially, like if you're going to map these two things back and forth. So here, if I say the same color on the natural image manifold of green, for example, would match the same color green in the latent space.

So now, let's visualize what happens as this gets trained. So here, if we're watching this thing get trained, you can see how it's going from the natural image manifold to that latent space. It's almost like you've taken something and tried to crumple it into a ball. And so even though we've said explicitly that, OK, everything in that latent space should correspond to something on the natural image manifold, actually, there's this weird stuff here.

There's danger zones where essentially, here, it's going to be closer to the natural image manifold. But if you're going from here to here, there's going to maybe be seams in that linear interpolation that will correspond to pretty jarring shifts where things that are nearby in latent space could potentially be quite far away from each other in data space. And so essentially, that z representation, if you're doing linear interpolation across that seam, it could be unnatural.

And that might be one of the ways to motivate the discovery that they found in the StyleGAN paper, which is essentially that using some intermediate representation somewhere in between that's not quite fully convex, but maybe like a little bit less distorted-- so some kind of intermediate thing where you're possibly less likely to jump across a seam or fall into some danger zone.

Basically, just somewhere in between might be the right place to interpolate and get things that look more realistic and behave in a more friendly way. And empirically and qualitatively, that's sometimes true. It's our favorite way to say things, right? Sometimes.

But it does actually correspond in some nice ways. So here, this was a work where they took that w space, that intermediate space, and they demonstrated that they could use it using-- this was a work called StyleSpace-- to very cleanly isolate and disentangle these dimensions of variation. And they got some really nice qualitative results.

So here, they're able to vary things at the level of granularity of hood styles or headlight types or body colors or background. Yeah. So it does seem to suggest that basically by removing some of those, like weird, badly behaved portions of linear interpolation, by moving to this more intermediate space, is beneficial when you're trying to find these disentangled dimensions of variation.

Cool. So now, let's talk about how you would label this generated data. So you have a label for all your real images. Well, maybe you don't, but assume you do. Assume that the data you started from, that you're training this generative model on, you had labeled. And now, you have some space in here that corresponds to that real data. But then there's a lot of other space that corresponds to purely generated data. So the argument here is that by construction, that things that are close together in this manifold, should be semantically related.

And so if you don't move too far on the manifold, you should argue that the category should stay the same, or at least this is the assumption. And so then you can rely on the normal inductive bias and machine learning that assigns similar labels to nearby points, if it's reasonably well semantically clustered. So here, if you know, for example, these are real data points where you know their true label, then you could assume that things sampled from those similar data points would actually correspond to similar things.

And so here, maybe in this region, yes, there's dimensions of variation that are captured, but they're not changing the underlying semantic meaning of the object itself, that these small variations are more corresponding to these dimensions of variation in the context, in maybe the scene, the visualization, but it's not changing the category that's being captured.

And so this was explicitly explored in this paper called DatasetGAN where they basically trained a GAN on one type of data, and then they wanted to solve a new task related to that data, so specifically labeling the parts of a car with semantic segmentation. And so then what they did is they used StyleGAN to efficiently label data in terms of the semantic segmentation labels, which can be quite expensive, by defining an additional labeling arm, which is essentially the output of a StyleGAN with just a few examples.

And then you can use that model to generate a really large data set of images and corresponding labels. They are weak labels, but there are labels that are explicitly based on this well-structured, well-aligned latent space. And then you can train a part segmentation model on the synthetic data and test it on the real data and show that it's actually beneficial.

So being a little bit more specific about this, essentially, if you're trying to train that labeling arm, they do that using the latent space and then just a really, really small number of manually labeled examples. And this is valuable, particularly because semantic segmentation labels are really expensive. How many of you have ever actually tried to label data for semantic segmentation?

Yeah. It sucks. It's super slow. Especially the parts that are really fine grained, like the boundaries of objects, it just takes forever to really try to get it very correct. And so being able to efficiently benefit from just a few of those and then be able to generalize well is pretty valuable.

So essentially, here, you map human labels in the pixel space through the network to some of these intermediate features that have some sort of maintained spatial relationships and arrangement. So basically, you take the original data, and then you run it through what they're calling a style interpreter, but it's essentially something like a StyleGAN.

That gets you a representation of these specific pixel vectors that you can then train a model, a really efficient model, predictive model, to map between those features and part labels. So essentially, it's like relying on the fact that training this generator forced the model to learn a really well-behaved, well-aligned representation space in terms of these dimensions of variation and similarity.

And then they could show that you could really efficiently train these accurate predictive models because the features themselves were better organized and more semantically meaningful than the pixels that you might be starting with. So instead of trying to go from images and pixel labels to a trained model that generates pixel labels given this input image, you basically take the input image, run it through to some intermediate feature space, and then you've trained this really lightweight predictive model to go from those features that are learned by a generative model to the correct semantic segmentation.

And they had some pretty nice examples of how they were kind of able to do this. And qualitatively and quantitatively, they were able to show that this worked quite well, and they got this kind of nice result, which is that a single labeled GAN image was worth about 100 labeled regular images, just in terms of the training efficiency versus the same amount of accuracy.

And these types of things are increasingly commonly explored, particularly for these types of labels that are really expensive to collect. And interestingly, this question of, how do you efficiently train these types of things based on well-learned representation spaces or foundation models, even ones that are not necessarily generative? has also been really well explored.

And so these days, if you're trying to do semantic segmentation, most people start from something that's called SAM, the Segment Anything Model, which is a model that relied essentially on building a really large data set using some pretty cute hacks for self-supervision.

You take an object, and you use copy paste augmentation to guarantee that you have that correct mask now in a bunch of different relationships and orientations. You use some similar ideas from something like DINOv2 that really captures fine grained semantic meaning in those feature spaces, and then you're able to learn a really generalizable and robust segmentation model on top of that.

Cool. So then you can also think about how these generative models' data plus plus approach can start to teach you how to explain things or increase the interpretability of some of these dimensions. So in standard classification, we take an image, we run it through a classification model, and it tries to predict cat. It's like, OK. It tries to predict a category.

But now, the question is, can we say something interesting about why the image was classified as a cat using this ability and this data plus plus space to explore counterfactuals? So here, you take the cat. You take it from this data space and turn it into this quote, unquote, data plus plus space where now you have some maybe encoder, some generator.

So you're mapping through, essentially, your G inverse and then through your G to get the same data. But now what you can do is you can look from this predicted class using-- from the generator, looking at the classes that are predicted, you can now explicitly try to perturb in the latent space to understand how that would cause your prediction to change.

So this gives you some ranking of the changes in the latent space and manipulations of these latent variables that maximally lead to the idea of changing a concept. So it helps you understand what maybe features of this image are most identifiable as the category of interest. And so here, this StylEx, it's like almost like StyleSpace explanation approach. It tries to find the topK StyleSpace directions, so these latent space directions of variation, that will most affect which class the model predicts.

And so here, there's some interesting stuff, right? You can look at what happens if you open the mouth of the thing, and it turns out that that is a really strong dimension of variation for moving the prediction from cat to dog, probably because dogs tend to pant a lot more than cats do. So it's a very cute interpolation.

You can also look at changing something about the size of the pupils relative to the eyes, and that also shifts the probability that you have something like a cat. Or you can think about changing how pointy the ears are, and that also tends to correspond from an explainability standpoint to why the model thinks it's a cat versus something else. And so this kind of gives us this sense of-- it's just another mechanism to try to probe the underlying things that the model has learned or has understood.

And you can imagine that this does tell you, again, something about the underlying bias in our training data, because there are many cats that have ears that are kind of shorter like that. British shorthairs have short ears just like that. But more often than not, something that has long, pointy ears is a cat, and maybe something with a shorter or folded over ears might be a dog. So it's capturing some dimensions of bias, I think, in the data as well.

And so you can think about this in terms of how we generate maybe class specific explanations. So here, if you have something like a perceived age classifier, then you can start really probing these weird dimensions of bias and see how that affects the perceived age of a model. So now, the model is predicting an age.

And so one of the things that's quite odd here is you'll see specifically that thicker eyebrows are corresponding responding to more youthful age. Lighter skin is corresponding to more youthful age, though also you'll note that it's lighter, but also sometimes less textured, which is one of these dimensions.

It's not a super, super disentangled dimension of variation. Or something like adding glasses or very obviously going to gray hair. So these were all things that were learned in terms of finding these dimensions of variation that corresponded to specific types of prediction. And it maybe makes sense.

Older people might be, in terms of your data bias, more likely to have white hair. That's something that's actually pretty ubiquitous. But you could also imagine that some of these dimensions of variation would not be something you would want a model to necessarily learn. These might capture dimensions of bias that may be related to the fairness of the equity of a model that we deploy.

So then what do you actually do with what you learn? Well, one way we could look at this is like, well, instead of looking at things like these dimensions of variation from internet images, can we take it to scientific data or medical data? So here, this is using that same type of explainability mechanism from a generative model to try to understand how to categorize different retinal fundus images.

And here, we're showing the top four examples of things that real doctors are looking for and then how you might actually change that image maximally to correspond to something predictive of a specific disease, for example. And this is quite interesting because it does actually-- if you show these to a doctor, it does correspond with the types of even very fine grained features that a doctor might use to actually categorize a specific type of disease.

So from that sense, from an explainability standpoint, it's almost a way to build trust in a model because an expert would say, OK, yeah. No, I do agree. I do agree that is a reasonable dimension of variation that I also would correspond to a specific type of disease. It's almost a reassurance that the model isn't picking up too many odd other correlative factors that might be things that we don't want to model to correspond to.

But of course, one of the really big limitations of all of these approaches is that you are required to figure out how-- you have to discover those latent space variations and then manually analyze or to find what they might correspond to. And that takes a lot of effort, right? So getting to the point where they had these examples of these specific dimensions of variation that corresponded really well to these real diseases, I'm sure it took a lot of handcrafting and a lot of time.

And so recent work tries to do similar types of counterfactual reasoning or understanding. But here, I'm actually just using text conditioning for modern diffusion models, or other types of conditioning using diffusion models, that give us really simple and interpretable mechanisms for control, though, of course, there's not always perfect alignment between maybe your text control and what actually gets generated by the generator.

But it does make this ability to test these counterfactual hypotheses, in many ways, pretty efficient. So I showed this work actually before, but this is work where they very specifically looked at using almost like human intelligence to generate possible counterfactuals, using text conditioning on generative images, and then explicitly testing, quantitatively, the performance of models given these different human-derived counterfactuals.

And they showed that it's just a really nice way to do, essentially, interfaces with your data sets, basically using the generator trained on top of data as a direct investigation mechanism or interaction mechanism with the underlying real data set itself understanding the dimensions of bias.

Cool. And then I think there's also this question of-- OK. The data the generated data is potentially useful for explainability or for our data set exploration. But it is also useful for representation learning? Can you learn from generated data in the same way that you might be able to learn, or maybe even in a better way than you might be able to learn, from real data?

And this is where it gets kind of complicated. So here's a work where they basically said, OK, you have some data set x, and then you're going to train a generative model on top of that data set. And the idea is somehow the generative model trained on top of the data set is going to capture either the same information as the data set, or maybe arguably slightly more information than the underlying data set.

And then that might actually be able to be useful for representation learning. So here, kind of similar to that other example with the multi-view ensembling from GANs, they're explicitly looking at your ability to generate maybe the same real image from different perspectives or from different views, and then use that as the input to contrastive learning models. So here, trying to learn a representation space where you're saying, all right. Here's two images from the same category. Here's one from a different category. And then you're training that contrastive style loss. Yeah.

AUDIENCE: So is this trained from scratch, or is this transfer learning to some degree?

SARA BEERY: Yeah. That's an interesting way-- it's an interesting question. I mean, arguably, this is a mechanism of transfer learning, assuming that--

AUDIENCE: Well, I guess to get to the original model, was that built on some pretrained model for image analysis?

SARA BEERY: Which original model? We're talking about kind of a system here.

AUDIENCE: [INAUDIBLE]

SARA BEERY: This generative model? So let's assume for the sake of simplicity that this is trained from scratch on that, because otherwise, it gets even more complicated to understand. And this actually, I think, is one of the difficulties currently when people are trying to understand whether we can-- there's this kind of broader meta argument going on in the community of whether it is possible to get more information out of a generator trained on data than the underlying data itself.

From an information theory perspective, it's like, how could you possibly create something from nothing? If you have the data, and that's all the information you have, you couldn't maybe get more information out of it. But then there's kind of the counterargument, which is actually like our architecture design, and the design of the training systems builds in knowledge that would not necessarily have already been built in.

And actually, you could argue that the construction of things like convolutional neural networks fundamentally are an injection of additional knowledge beyond just the underlying intermediate data. We're building knowledge into the design of our architectures and our loss functions based on our understanding of what, for example, structure we expect to see. So convolutional neural networks basically say, we expect there to be local structure in images.

And so then the argument is, isn't data augmentation a mechanism-- you could argue that random flipping is an image generation algorithm, right? I mean, depending on how you want to define that, semantically, it is, right? You've taken an image. You've used some algorithm to generate a new image that you guarantee still has fidelity to the class that you care about.

And then things like this copy paste, cut and paste style data augmentation, that's even getting maybe closer to something like a learned generative model, because now, you're building explicit assumptions about taking foreground objects and putting them on different backgrounds.

This is like a really useful mechanism for increasing the robustness of machine learning models, just these engineering hacks about how we manipulate the data during training. And then maybe the argument is like, how is that any different from something like learning a generator that can interpolate maybe in something closer to the image manifold? So it's a bit semantic.

Anyway, I have been thinking about this a lot. I have a grad student who's been looking at whether we can-- I really, really specifically want to be able to learn, maybe using generative models, to do a better job, a more robust job, of recognizing rare things. But now, you've really gotten yourself in a chicken and egg problem.

Because the thing is rare, it's hard to train a generator that does a good job of generating it, and then the reverse is also true. And there have been people that have shown that you can improve rare categorization using generative models, but they cheat, and they train the generative models on a lot of data. So far, we have been able to find ways where you can get some gains in very specific constrained scenarios, but it's not just naive and simple.

Anyway, the point here is that you can take the classical contrastive learning approach where we're using our knowledge-based image generation, where we've done these specific types of warping or whatever, random cropping, flipping, to the initial image color jitter, in a way that we know doesn't necessarily break the category boundary. We still want it to be the same thing.

And here, we're instead going to move just a little bit in some latent space to create now this mapping in data space through what should be the manifold of real images, and then use that as your transformation for your contrastive learning model. Yeah.

AUDIENCE: I had a question about this, which kind of relates back to earlier when we talked about the idea of small local movement in the latent space not crossing a boundary. I mean, intuitively, if you have just a 2D latent space and a 10 class classification problem, you could imagine some very naive pie chart that has 10 slices. And there has to exist a boundary somewhere between category 1 and category 2.

SARA BEERY: Yes.

AUDIENCE: So is this just a probabilistic argument that in a high enough dimensional latent space, the odds that you would sample a location where the boundary shifts is so unlikely that you can get away with this kind of approach to training for contrastive learning?

SARA BEERY: Or I would even go beyond that, and I would even say as long as you define some dimension of movement-- as long as, most of the time, it's not crossing a class boundary-- if it does cross a class boundary a very small amount of the time still, by the way we train these models, it's almost like the noise will come out. It's almost a mechanism for regularization.

Of course, there's counterarguments to that. It depends on how egregious those boundaries are and how frequently you're crossing them. Yeah. I would say as long as-- if you want to make a probabilistic argument, and you want to map that to something that's like a physical area argument, like the area of the model where moving in a small amount corresponds to a reasonably maintaining the category boundary relative to the areas where you are crossing category boundaries-- as long as that ratio is quite large, then I imagine you would still learned something useful, even if sometimes it's wrong.

It's just like how in the standard contrastive learning objectives, we did see some improvement when you could explicitly remove these false positive, false negative type things where, within your batch, you would have maybe two images of dogs. And now, they're being treated as a different category during your representation learning. You're explicitly saying those things should be far apart, though they actually represent the same category.

If you could remove all of those, the models did train better. There were these nice examples where you could show, if you removed these kind of false characterizations by using supervision, for example, you got better representation learning. But the model's still able to learn useful representations even when those do exist in there. Yeah.

Cool. So yeah. Now, here, it's just a different mechanism for constructing these positive pairs to train your contrastive learning algorithm. And so here, as opposed to your standard SimCLR type views where you have two views of the same thing where it's maybe cropped from different areas or warped in terms of the color, now, you're taking latent views. You're taking some ball around the real image. And any dimension, you would move in that ball, you could argue, would be a different view of the same thing.

And then now, just how we might say, OK, these are all different views of the same thing for the purposes of contrastive learning. Here, you can say, these are all different views of the same thing. Now, one interesting dimension here, just generally, is this question of diversity versus fidelity when it comes to the usefulness of training signal.

This is also something I've been thinking about a lot. So it's probably more useful to know that two things that are more different from each other are, in fact, the same than it is to know that two things that are almost identical to each other are, in fact, the same. So for example, if you look at this middle set here, where these are all American robins, the pose of this American robin in basically all of these is nearly identical.

The construction of the image, the orientation, et cetera, of the image is almost identical. There's not a lot of diversity here. And so actually from a learning signal perspective, this might be less useful, arguably, than something that also does something like random flipping. And so I think there is this kind of interesting challenge with these generative models, broadly, with GANs, with diffusion models, et cetera.

Even with some of these personalization style models that are trying to do a really nice job of capturing something specific, often, there's almost like a Pareto frontier with these models when it comes to this trade off between diversity and fidelity. And so you can't to get the model to generate things that are really diverse, maybe in terms of their context, or their scene structure, without giving up fidelity.

You end up in this space where you can generate a robin that's positioned differently in the scene, maybe flying, different pose, and you can try to force the model to do that. And it will, but it probably won't actually be very robin-like anymore if this is the input image that you're working from. Yeah.

AUDIENCE: In the latent space, when you're moving around the image feature vector, would you say that the images around it belong to the same class? How do you know how far you can go? And that distance, is it the same for all classes? You can imagine for classes of one, one class is overrepresented compared to another class.

SARA BEERY: Yeah. So the question is basically, how do you know how far you can go in the latent space and have it still map to the same category? And does that distance correspond-- is the optimal amount of distance you could go for any given category the same across all categories? So I would say experimentally, in a lot of these papers, they're using this as a hyperparameter, right?

They're basically saying, all right. We're going to define some radius in something like cosine distance, and we're going to sample within that radius. And we're going to test for different radiuses or radii. And then we're going to pick the one that does the best on our test data.

That's a bit disappointing, because I actually think that's a really interesting question. How do you determine how far you can go before it crosses a class boundary? And almost by definition, you start getting into these really messy questions, particularly as these models are able to generate things that are realistic looking but impossible.

Me and Phil actually argue about this quite a lot. If you had a picture of your mom, but you had another eye in the middle of her forehead, is it still your mom? What do you guys think? Is it still your mom if she has three eyes? There's not a right answer here.

And so if some of the dimensions of interpolation you can move into are still kind of realistic looking, but actually impossible-- so maybe if you think about this from a species categorization perspective, you take like this American robin, and then you make the chest gray instead of red. That's not a real bird. So now, you have things that look like birds, but they don't actually correspond to any real category of bird.

And so then what should that be categorized as? And so somehow, understanding the boundary of a class, a really important component to being able to do that is having representative data from all the dimensions of real variation for that class so you can define what the reasonable class boundary is.

That's one of the reasons that few shot learning is so hard, because it often means we don't have a good, well understood representation space of the boundaries of the real class. And so we don't know. If you've never seen a picture of a bird as a juvenile, we don't know what the real class boundary should be. Yeah.

AUDIENCE: Isn't that just like a fundamental shortcoming of classification as a learning objective? CLIP was almost a better idea where they're just saying that you should learn to embed text descriptions of images close to that embedding of the image, and you can capture more truth about the world. To your point about the robin chest-- I mean, this isn't exact.

But imagine you have an albino robin. That's still a robin, and a human can look at it and be like, yeah, that's a robin. But from your definition of the classification perspective, I doubt-- I think you're right that for a specific type of classification, whatever you define that class to be is the extent to which the model can learn [INAUDIBLE]. So isn't this just a shortcoming of classification?

SARA BEERY: Yeah. So the argument is, basically, this challenge of being able to understand category boundaries is maybe a limitation of the underlying philosophical concept of trying to categorize things with some sort of list of specific categories.

And actually, if you really want to get deep into it, there's a lot of literature from philosophy about the concept of categorization. And there's kind of these explicit differences of opinion when you look at historical philosophers around the idea that it's possible to categorize something as a list of all its attributes.

So you could say, OK. A robin is a bird with a black face and a red chest and yellow feet and a round structure and a yellow eye with a black pupil. And so there's this idea that any object can be defined as a list of characteristics you're looking for. And then if you have the kind of mapping of the set of characteristics to the things you care about, then that's all you need.

My argument is kind of like, that's the CLIP approach, in a way, because you're trying to learn something about the relationship between textual semantic concepts and things in images. And there's been a lot of work since CLIP that actually tries to really explicitly ground semantic concepts from text into the visual components that corresponds to.

But I think the argument is that, no matter what, if you give me a list of a fixed list of attributes, and you try to map that to a category, you can basically come up with a counterexample that breaks it. And then there's the question of, where do you want to draw the boundaries? That somehow has to be related to some goal. So I'm getting really philosophical here.

So if I showed you like an apple and a bell pepper, and there was a green apple and bell pepper and a red apple and bell pepper, there's multiple reasonable ways to split the set of groups. So in the end, it always just comes down to, what do you really need the representation space to do for you? And then what types of dimensions of variation do you want to capture there?

And then the albino robin example is a great example of ways that we are very good at generalizing, but I would actually argue that models like CLIP still cannot capture effectively. Yeah. And then you could tell me, if you were a bird expert, oh, actually, that looks like it might be an albino robin, but that's actually this other species. And I'd believe you, because I don't know.

Yeah. So I think this is really fascinating. I think it starts getting at some really interesting kind of meaty questions about, what do we really want from representations? And what does it really mean to move through a representation space?

But so here, again, kind of going back to this idea of, do contrastive learning from real data based on some of heuristic set of rules that we say don't break the class boundary versus if you do it from generative data based on some experimental setup of parameters around how far you can move in any given direction, that you're able to now, instead of going from some data space to a representation space, go from some latent space into a data space into a representation space.

And they were able to show that if you're doing linear transfer from ImageNet1000, you can get 43.9% top-1 accuracy here. But only using real data to train these models now mapping through these kind of information bottlenecks in a way of these different types of latent spaces, you can do pretty well. You can get 35.7% accuracy. But then if you add in that ability to move in that latent space, the accuracy goes up to about 42.6%.

So here, you'll note neither of these are as good as just training on the real data themselves. So clearly, there's some limitations in terms of the mechanism. But the argument is that if you were able to build the system effectively, the ability to interpolate and have these reasonable dimensions of variation within that original latent space could potentially add a lot of value. This is a pretty significant boost in performance.

So then the argument is that these deep generative views is one dimension of variation, just different viewpoints, can improve contrastive learning beyond only using standard data augmentation. And if the generative model is high quality, you could even potentially outperform learning from real data. But that, of course, means that we need the generator to be high quality, which it's not for many things that are uncommon.

One of the things that is kind of fun, if you try and play with these models, is looking at what they understand and trying to get them to break out of their priors. So there's some classic examples of this. If you ask models to generate a bunch of pictures of birds, if you ask it to generate orioles, it'll give you baseball players, right? There's this semantic prior that gets broken by the training data in really interesting ways.

But then there's also other interesting stuff. If you ask the model to generate a toucan with a short beak, it's really hard to get it to do it. It just won't. It's like, no, no, no. Toucan somehow overpowers these mechanisms of morphology. And I've also found that they're much better at generating dimensions-- they're much better at changing texture than shape.

So it turns out that the models seem to have learned really strong shape priors that are hard to break out of for any given category, but they've gotten quite good at swapping textures for a category. So you can say, oh, make a baseball, but make it furry. And it's like, OK. It swaps the texture just fine. But if you say make a baseball, but make it star shaped, it gets a little harder.

And I think that's really interesting. I think it's interesting to think about why shape might be encoded in a stronger way than texture. And there's some recent work that shows some really nice-- in this space of trying to build effective data augmentation through generation. It just came out at ECCV. They basically use the scene structure as a fixed thing.

So they mask out the entire dimension of the scene. So now, you're guaranteeing that you're going to have a realistic scene because it's a real one. So maybe you have two dogs sitting on a bench. Now, the way that they actually do their generation to get these new cases is they take each individual object, and then they change it to a different instance within the data set.

So you basically say, I'm going to learn some personalization model for every individual dog in the data set, and then I'm going to condition on the mask that I have and generate it with that instance. So they basically call it instance augmented generation, and they do get some really nice results. And what it is, basically, it's breaking these correlations between scene structure and individual objects in a really realistic way.

And the images do look reasonably correct. But the shape thing gets weird again. So because that data set has bird defined as a category, you can take any bird of any shape, and you can replace it with the texture, essentially, of a different bird. But birds are very different shapes, so it'll take something like a little round bird, and then it'll try to squash an eagle into that shape.

And this is something that wouldn't happen if the granularity of their categories in the labeled space was a little bit more fine grained. And again, it's just built in assumptions about the amount of shape variation you might have within a given defined category of interest from a human perspective. Anyway, that's enough philosophy for now.

So essentially, data plus plus says that you might be able to sample from an implicit generative model and use it to act like some of decorative or interactive or mechanism to get data with extra functionality. And so you can define operators on that, things like interpolation, extrapolation, manipulation, composition, optimization, and you can use it to label data potentially quite efficiently.

So the argument here is that maybe everything you can do with regular data, you could also try to do with this generation-augmented data. And it might work better. But I think there's still a lot of research that needs to be done in this space to really achieve the benefits that we think there's potential for. We need generators that can learn with less data, and we need to better understand how these things are being learned. Yeah.

AUDIENCE: So two questions. One on the bias. So I'm just curious to know, how can we figure out if there's biases in the training? Like in this case, we correlate darkness equals volcano eruption without seeing the images, because this is-- we can see it visually, but if the data set is so large we can't see it.

And the second one is, how do we prioritize diversity versus accuracy? So the example you gave of a little bird with the eagle features, but it's the size of little bird. Is it great because it allows me to train for things that aren't even possible, so maybe my testing data will improve? But it's so far away from fidelity. Does that make this driven data bad?

SARA BEERY: Yeah. So it's kind of like that question about the third eye in the middle of the forehead. If you start adding random eyes to your mom, and it makes the model more robust at recognizing your mom in real data, is that a problem?

So with relation to the first thing, for example, that data set interfaces paper was a mechanism that tries to use the fact that you can interact with these systems with text to probe biases efficiently without needing to explicitly look through all the data. So I think there's ways people have thought of using generative models to help you find biases efficiently.

And then with respect to the second one, I think it basically depends. There's some work that tries to demonstrate that even for really fine grained things like bird species, you can learn to generate data that is accurately of that bird species by using a classification model as guidance for the generator.

But essentially, what that means is then you're building a generator that's optimized for a given classification model, and you can take that to its logical extreme, because it essentially means you can map out the boundaries of that classification model really explicitly moving into spaces that aren't super realistic. And then you can maybe get better performance for a given manifold or a given representation space.

But it might not always correspond to things you want. So for example, you get better performance on crows when you add in images where it's like Bigfoot with a crow texture. And it kind of gets to this weird question of, we're getting better results on our test data, but is it actually going to correspond to better results in practice beyond this of fixed set of test data?

And I think that gets kind of hard to quantify. So we're good at optimizing for metrics on test data sets, but I think it's much harder if you're moving to the real world. You could imagine it might actually be better in some ways to have things that aren't, I guess, different in ways that-- weird and not realistic. Because maybe in the real world, you might be more likely to encounter something that actually does look a little bit like a small eagle, but it's actually something else entirely.

So maybe the open endedness of the real world makes the question of maximizing the potential within a kind of a fixed, closed world setting a little bit less translatable. But yeah. I don't know if that kind of makes sense. I'm sort of ranting. But I do think that what is useful is going to be directly related to your task. So for different types of tasks, for different goals, useful data will be different. Yeah.

AUDIENCE: Is image domain the bulk of where the cutting edge research is being done for this topic, or is that a pedagogical reason?

SARA BEERY: Why am I showing a bunch of image examples?

AUDIENCE: [INAUDIBLE]. It would probably be harder to do these evaluations on text because you'd have to each read each one and see the differences.

SARA BEERY: Yeah. That's interesting. So there definitely is research on generative text as data augmentation, but I think you're right that it's kind of harder-- class boundaries and text is a little bit harder to define. And also, I could imagine if you think about generative songs, for example, maybe part of it is just the nuanced, messy space of what a category is.

But there definitely is work on both generating data and using generative data for training in the tech space and in the audio space and in many other modalities as well. Lots of work. Actually, classical work that used generated data for training in medical imagery, because often, we had so few real data that really carefully handcrafted synthesizers of data were quite effective.

There is kind of a much broader kind of interesting debate currently about what's called model collapse or manifold collapse where, essentially, if you start training a model on more data that's generated than data that's real-- which is a very clear concern today, because more and more of the text on the internet is generated text as opposed to real text-- there's an entire line of research that essentially shows that once you go too far down the ratio of more of the data being generated than real, it becomes kind of a vicious cycle.

And you can have these types of model collapse where you start to lose parts of your distribution that were captured in the real data because of the kind of distribution of the generated data based on the likelihood. And there's fascinating-- it's a pretty active area of research right now, this question of, what does it look like to have models trained from generated data? And what are the kind of failure modes that can result in?

Cool. So in the last little bit, I'm going to talk a little bit about learning to learn quickly. So this idea of not just, what is a mechanism for transfer learning? But actually, how do you learn a model that is good at transfer learning, a model that adapts quickly?

So here, the idea is that maybe instead of just taking the classical approach where you have learning, which is this middle part where you have a data set, a learner, you get some parameters, and then at inference time, you take your data set, you run it through a model with parameters, and you get predictions, you add this meta layer on top of taking a bunch of different data sets and training some of meta learner that has a different set of hyperparameters that then you transfer to any given learner and data set.

And so one of the seminal works in this space was this work called MAML by Chelsea Finn-- stands for model-agnostic meta-learning. And so far, we saw transfer learning as, given some prior model and representation, how can we quickly adapt it? And this was the first work that tried to ask, could we learn to do transfer learning effectively? So now, consider you're doing fine tuning. Given some initialization we, update it with SGD. MAML tries to learn an initialization such that it's easily adaptable with just a few steps of stochastic gradient descent.

And so if you look at this algorithmically, essentially, you now are doing like a double loop in your optimization. So starting with your initialization, for any given data set that you're going to train over, you now run a few steps of SGD and then compute loss after those steps. And then you backprop through that back to your initial meta-learning initial set of parameters.

So how do you actually backprop through something that's now like an inner loop within your larger loop? So if you're doing one step of stochastic gradient descent, you essentially can consider that a forwards pass over x plus the backwards of the forwards of x and theta plus theta. Essentially, remember when we took a single part of a model, and then we showed that if you wanted to do back propagation, you could basically unfold it, and it was essentially one big forward pass of going through forward and then backward of a model?

So now if you want to do two steps of stochastic gradient descent, again, you're kind of nesting this. And so essentially, this is saying that you can backprop through backward. So if you're going forward and backward through a simple network-- I don't know if you guys remember this. So here's that forward pass. Then, you have your backward pass.

And then here, this is where we kind of unfold it. And so now, you can show that doing forward and backward through a simple network is just one forward pass. And then you can do this meta backward through everything. So it's possible. It's just expensive, essentially, is the main point here, because you're basically taking those nested loops of forward and backward, and you're just stretching all of them out.

And you're now getting what is a very, very deep network that you're going to optimize through. Now, this type of computational complexity often means that the relative benefit of these types of meta-learning models-- though nice in theory, and you get really nice gains on maybe simple problems-- it's often just too computationally expensive to really deploy.

But there are a few examples of how people have actually used these types of meta-learning for real world applications. So in this case, in practice, I was saying the gains of the method, something like model-agnostic meta-learning, are often not worth the additional computational cost at training time, and sometimes can be worse than just like pretraining on all of those different data sets and then fine tuning.

But Gabi Tseng introduced this geospatial embedding, essentially like a bit of a task-specific tokenizer, within that meta-learning objective that encoded some kind of task-specific context. And they showed that giving this task specificity within that meta-learning loop did give them some really nice gains when they were looking at crop type categorization for corn and a bunch of different countries.

So here, these are a comparison of a bunch of different types of methods. But you can see at the top is kind of versus meta-learning or just pretraining a model, there, it's about the same. But this task conditioning that gives your model the ability to specialize effectively, maybe to a given country or part of the world, ends up being pretty effective.

This is maybe-- I don't know if you guys remember, but we talked a bit about this idea of maybe learning a prior when we were talking about distribution shift. This is maybe a different mechanism of trying to learn something like a geospatial prior or some of useful contextualization so that your model can actually build some knowledge about how it should adapt for specific types of problems or types of tasks.

And so there's also this question of meta-learning by sequence modeling. So definition of learning. Maybe we have a sequence of input and ground truth, input and ground truth for a bunch of things. And then you output some function such that if you give it a new input, it'll give you a new ground truth.

So maybe this is kind of like that example we talked about when we were talking about autoregressive modeling for large language models where we could say, OK. We're going to show you an example of English and the translation in Spanish. I'm going to show you another example in English, another example of the translation in Spanish. Now, if I show you something in English, the model would automatically generate the translation in Spanish. This idea of sequence based in-context learning.

And so here, if you have something like a recurrent neural network, the state plus the parameters of that recurrent neural network, or something like an autoregressive model, like predicting the next word, is f, that current state. And then you apply that f to your new input. You're basically saying, predict the next thing. And so then the hidden state of that RNN or that recurrent neural network is a model that arguably learns to learn, right?

It takes some statistics of the sequence in general that enable it to predict what should come next. And this really is, I think, what GPT and a lot of these modern autoregressive models are doing. It's kind of a version of meta-learning at large scale. And you could argue that in-context learning is kind of just another mechanism of meta-learning. Because when these models are being trained, you're showing them a lot of different possible tasks or sequences and then asking them to learn how to really quickly and efficiently generate answers.

What comes next for new sequences? So you could argue that there's a pretty similar motivation and potentially a much better actual output when you think about meta-learning through sequence modeling as opposed to meta-learning through this kind of inner and outer loop of training.

Cool. So it's the end of the lecture I had for today. We learned today about generative models as data and some of the pros and cons of that, and we also talked about learning to learn, or meta-learning. And like I said, it's my last lecture of the course today. So it's been really fun teaching you all, and hopefully-- I mean, I'll be around.

[APPLAUSE]