ADAM YALA: OK, great. Well, thank you for the great setup.

So for this section, I'm gonna talk about some of our work in interpreting mammograms for cancer. Specifically it's going to go into cancer detection and triage mammograms. Next, we'll talk about our technical approach in breast cancer risk. And then finally close up in the many, many different ways to mess up and the way things can go wrong, and how does it [INAUDIBLE] clinical implementation.

So let's kind of look more closely at the numbers of the actual breast cancer screening workflow. So as Connie already said, you might see something like 1,000 patients. All them take mammograms. Of that 1,000, on average maybe 100 they called back for additional imaging. Of that 100, something like 20 will get biopsied. And you end up with maybe five or six diagnoses of breast cancer.

So one very clear thing you see about problems when you look at this funnel is that way over 99% of people that you see in a given day are cancer-free. So your actual incidence is very low.

And so there's kind of a natural question that can come up. What can you do in terms of modeling if you have an even OK cancer detection model to raise the incidence of this population but automatically reading a portion of the population is healthy. Does everybody just follow that broad idea? OK. That's enough head nods.

So the broad idea here is you're going to train the cancer detection model to try to find cancer as well as we can. Given that, we're going to try to say, what's a threshold on a development set such that we can kind of say below the threshold no one has cancer. And if we use that at test times, simulating clinical implementation, what would that look like? And can we actually do better by doing this kind of process?

And the kind of broad plan of how I'm gonna talk about this-- I'm gonna do this for the next product as well. Of course, we're going to talk about the kind of dataset collection and how we think about, like, you know, what is good data and how do we think about that.

Next, the actual methodology and go into the general challenges when you're modeling mammograms for any computer mission tasks, specifically in cancer, and also, obviously, risk.

And lastly, how we thought about the analysis and some kind of objectives there.

So to kind of dive into it, we took consecutive mammograms. I'll get back into this later. This is actually quite important.

We took consecutive mammograms from 2009 to 2016. This started off with about 280,000 cancers. And once we kind of filtered-- so at least one year follow up, we ended up with this final setting where we had 220,000 mammograms for training and about 26,000 for development and testing.

And the way we had it, it all comes to say, is this a positive mammogram or not? We didn't look at what cancers were caught by the radiologists. We'd say, you know, what was cancer that was found in any means within a year?

And where we looked to was through the radiology, EHR, and the Partners-- kind of five hospital registry. And there we were trying to save cancer-- if anyway we can tell a cancer occurred, let's mart it as such regardless of what others caught on MRI or some kind of later stage.

And so the thing we're trying to do here is just mimic the real world of what are we trying to catch cancer. And finally, important details we always split by patient so that your results aren't just memorizing this specific patient didn't have cancer. And so we have some overlap that's some bad bias to have.

OK. That's pretty simple. Now let's go into the modeling. There's going to kind of follow two chunks. One chunk is going to be on the kind of general challenges, and it's kind of shared between the variety of projects. And next is going to be kind of more specific analysis for this project.

So kind of a general question you might be asking, I have some image. I have some outcome. Obviously, this is just image classification. How is it different from ImageNet?

Well, it's quite similar. Most lessons are shared. But there are some key differences.

So I gave you two examples. One of them is a scene in my kitchen. Can anyone tell me what the object is? This is not a particularly hard question.

AUDIENCE: [Intermingled voices] Dog. Bear.

ADAM YALA: Right.

AUDIENCE: Dog.

ADAM YALA: It is almost all of those things. So that is my dog, the best dog. OK. So can anyone tell me, now that you've had some training with Connie, if this mammogram indicates cancer?

Well, it does. And this is unfair for a couple of reasons. Let's go into, like, why this is hard. It's unfair in part because you don't have the training. But it's actually a much harder signal to learn.

So first let's kind of delve into it. In this kind of task, the image is really huge. So you have something like a 3,200 by 2,600 pixel image. This is a single view of a breast. And in that, the actual cancer they're looking for might be 50 by 50 pixels. So intuitively your signal to noise ratio is very different.

Whereas an image that-- my dog is like the entire image. She's huge in real life and in that photo.

And the image itself is much smaller. So not only do you have much smaller images, but you're kind of, like, the relative size of the object in there is much larger.

To kind of further compound the difficulty, the pattern you're looking for inside the mammogram is really context-dependent. So if you saw that pattern somewhere else in the breast, it doesn't indicate the same thing. And so you really care about where in this kind of global context this comes out. And if you kind of take the mammogram at different times with different compressions, you would have this kind of non-rigid morphing of the image that's much more difficult to model. Whereas that's a more or less context-independent dog.

You see that kind of frame kind of anywhere, you know it's a dog. And so it's a much easier thing to learn in a traditional computer vision setting.

And so the core challenge here is that both the image is too big and too small. So if you're looking at just the number of cancers we have, the cancer might be less than 1% of the mammogram and about 0.7% of your images have cancers, even in this data set, which is from 2000 to 2016 MGH, a massive imaging center, in total across all of that, you will still have less than 2,000 cancers.

And this is super tiny compared to regular object classification data sets. And this is looking at over a million images if you look at all the four views of the exams. And at the same time, it's also too big.

So even if I downsample these images, I can only really fit three of them for a single GPU. And so this kind of limits the batch size I can work with. And whereas the kind of comparable, if I took just the regular image net size, I could fit batches of 128, easily happy days and do all this parallelization stuff, and it's just much easier to play with.

And finally, the actual data set itself is quite large. And so you have to do some-- there's nuisances to deal with in terms of, like, just setting up your server infrastructure to handle these massive data sets, also be able to train efficiently.

So you know, the core challenge here across all of these kind of tasks is, how do we make this model actually learn? The core problem is that our signal to noise ratio is quite low. So training ends up being quite unstable.

And there's a kind of a couple of simple levers you can play with. The first lever is often deep learning initialization.

Next, we're gonna talk about kind of the optimization or architecture choice and how this compares to what people often do in the community, including in a recent paper from yesterday.

And then finally, we're gonna talk about something more explicit for the triage idea and how we actually use this model once it's trained.

OK. So before I go into how we made these choices, I'm just going to say what we chose to give you context before I dive in. So we followed some image initialization. We use a relatively large batch size-ish of 24. And the way we do that is by taking 4 GPUs and just stepping a couple of times before doing an optimizer step.

So when you do a couple rounds of back prop first to accumulate those gradients before doing optimization. And you sample balanced batches of training time.

And for backbone architecture we use ResNet-18. It's just kind of, like, fairly standard.

OK. But as I said before, one of the first key decisions is how do you think about your

initialization? So this is a figure of ImageNet initialization versus random initialization. It's not any particular experiment. I've done this across many, many times. It's always like this. Where if you use image initialization, your loss drops immediately, both in train loss and development loss when you actually learn something. Whereas when you do random initialization, you kind of don't learn anything. And your loss kind of bounds around the top for a very long time before it finds some region where it quickly starts learning. And then it will plateau again for a long time before quickly start learning.

And to kind of give some context, to give about 50 epochs takes on the order of, like, 15, 16 hours. And so to wait long enough to even see if random initialization could perform as well is beyond my level of patience. It just takes too long, and I have other experiments to be running.

So this is more of an empirical observation that the image initialization learns immediately. And there's some kind of questions of why is this? Our theoretical understanding of this is not that strong. We have some intuitions of why this might be happening.

We don't think it's anything about this particular filter of this dog is really great for breast cancer. That's quite implausible. But if you look it into a lot of the earlier research in terms of the right kind of random initialization for things like revenue networks, a lot of focus was on does the activation pattern not blow up as you go further down the line.

One of the benefits of starting with the pre-trained network is that a lot of those kind of dynamics are already figured out for a specific task. And so shifting from that to other tasks has seemed to be not that challenging.

Another possible area of explanation is actually in a BatchNorm statistics. So if you remember, we can only fit three images per GPU. And the way the BatchNorm initialization is implemented across every deep learning library that I know of, it computes independently per GPU to minimize the kind of inter-GPU communication. And so it's also less able to kind of guess from scratch. But if you're starting with the BatchNorm statistics to ImageNet and just slowly shifting it over, it might also result in some stability benefits.

But in general, or like, a true deeper theoretical understanding, but as I said, it still eludes us. And it isn't something I can give too much conclusions about, unfortunately.

OK. So that's initialization. And if you don't get this right, kind of nothing works for a very long time. So if you're gonna start a project in this space, try this.

Next, another important decision that if you don't do, it kind of breaks, is your optimization/architecture choice. So as I said before, kind of a core problem in stability here is this idea that our just signal to noise ratio is really low. And so a very common approach throughout a lot of the prior work and things I actually have tried myself before is to say, OK, let's just break down this problem. We can train at a patch level first. We're going to take just subsets of a mammogram in this little bonding box, have it annotated for radiology findings like benign masses or calcification and things of that sort.

We're going to pre-train on that task to have this kind of pixel level prediction. And then once we're done with that, we're going to fine tune that initialized model across the entire image. So you kind of have this two-stage training procedure.

And actually, another paper that came out just yesterday does the exact same approach with some slightly different details. But one of the things we wanted to investigate is if you just-- oh, And the base architecture that's always used for this, there is quite a few valid options of things that just get reasonable performance and ImageNet, things like VGG, Wide ResNets and ResNets. In my experience, they all performed fairly similarly. So it's kind of a speed/benefit trade-off.

And there's an advantage to using fully convolutional architectures because if you have fully connected layers that are assumed specific dimensionality, you can convert them to convolutional layers. They're just more convenient to start with a full convolutional architecture. There's going to be resolution invariant.

Yes.

AUDIENCE: In the last slide when you do patches--

ADAM YALA: Yes.

AUDIENCE: How do you label every single patch? Are they just labeled with a global label? Or do you have to actually look and catch, and figure out what's happened?

ADAM YALA: So normally what you do is you have positive patches labeled. And then you randomly sample other patches. So from your annotation-- so, for example, a lot people do this on public data sets like the Florida DSM dataset that has some entries, of like, here are benign masses, benign calcs, malignant calcs, et cetera.

What people do then is take those annotations. They will randomly select other patches and say, if it's not there, it's negative. And I'm going to call it healthy.

And then they'll say if this bonding box overlaps with patch by some marginal call, it's the same label. So do this heuristically. And other data sets that are proprietary also kind of play with a similar trick. In general, they don't actually label every single pixel accordingly. But there's relatively minor differences in how people do this. But the results are fairly similar, regardless.

Yes.

AUDIENCE: When you go from the patch level to the full image, if I understand correctly, the architecture hasn't quite changed because it's just convolution is over a larger--

ADAM YALA: Exactly. So the end thing right before we do the prediction is normally-- ResNet, for example, does a global average pool. Channel lies across the entire feature map. And so they just-- for the patch level they take in an image that's 250 by 250, do the global average pool across that to make the prediction. And when they just go up to the full resolution image, now you're taking a global average pool over a 3,000 by 2,000.

AUDIENCE: And presumably there might be some scaling issue that you might need to adjust. Do you do any of that? Or are you just--

ADAM YALA: So you feed it in at the full resolution the entire time. So you just-- do you see what I mean? So you're taking a crop. So the resolution isn't changing. So the same filter map should be able to kind of scale accordingly.

But if you do things like average pooling, then you're kind of-- any one thing that has a very high activation will get averaged down lower. And so, for example, in our work, we use max pooling to kind of get around that. Any other questions?

But if this looks complicated, have no worries because we actually think it's totally unnecessary. And this is the next slide. So good for you.

So as I said before, this kind of, what are the problems that signal to noise? So one obvious thing to kind of think about is, like, OK. Maybe doing SGD with a batch size of three when the lesion is less than 1% of the image is a bad idea.

If I just take less noisy gradients by increasing my batch size, which means use more GPUs,

take more steps before doing the weight update, we actually find that the need to do this actually goes away completely. So these are experiments I did in the publicly available data set a while back while we were figuring this out.

If you take this kind of [INAUDIBLE] architecture and fine tune with a batch size of 2, 4, 10, 16, and compare that to just a one-stage training where you just do the [INAUDIBLE] beginning and initialized in ImageNet and as you use different batch sizes, you quickly start to close the gap on the development AUC.

And so for all the experiments that we do broadly we find that we actually get reasonably stable training by just using a batch size of 20 and above. And this kind of comes down to if you use a batch size of one, it's just particularly unstable.

In other details that we always sample the balanced batches. Cause otherwise you'd be sampling like, 20 batches before you see a single positive sample. You just don't learn anything.

Cool. So that is like, if you do that, you don't do anything complicated. You don't do any fancy cropping or anything of that sort, or like, dealing with like VGG annotations. We found that the actual using VGG annotation for this task is not actually helpful.

OK. No questions? Yes.

AUDIENCE: So with the larger batch sizing you don't use the magnified patches?

ADAM YALA: We don't. We just take the whole image from beginning. Pretend you-- like, can you just see the annotation as whole image, cancer with less than within a year. It's a much simpler setup.

AUDIENCE: I don't get. That's the same thing I thought you said you couldn't do for memory reasons.

ADAM YALA: Oh. So you just-- instead of-- so normally when you do, you're going to train the network, the most common approach is you do back prop and then step. Cause you do back prop several times, you're accumulating the gradients, at least in PyTorch. And then you can do step afterwards. So instead of doing the whole batch at one time, you just do it serially. So there you're just trading time for space.

The minimum, though, is you have to fit at least a single image per GPU. And in our case we can fit three. But to make this actually scale, we use four GPUs at a time.

Yes.

AUDIENCE: How much is the trade-off with time?

ADAM YALA: So if I'm gonna take one batch size any bigger, I would only do it in increments of let's say 12, because that's how much I can fit within my set of GPUs at the same time. But to control the size of the experiment you want to have the kind of the same number of gradient updates per experiment. So if I want to use a batch size of 48, so all my experiments, instead of taking about half a day, it takes about a day.

And so there's kind of, like, this natural trade-off as you go along. So one of the things I mentioned at the very end is we're considering some adversarial approach for something. And one of the annoying things about that is that if I have five discriminator steps, oh my god. My my experiment-- I'll take three days per experiment.

And [INAUDIBLE] update of someone that's trying to design a better model becomes really slow when the experiments start taking this long.

Yes.

AUDIENCE: So you said the annotations did not help with the training. Is that because the actual cancer itself is not really different from the dense tissue, and the location of that matters, and not the actual granularity of the-- what is the reason?

ADAM YALA: So in general when something doesn't help, there's always kind of like a possibility of two things. One thing is that the whole image signal kind of subsumes that smaller scale signal. Or there is a better way to do it I haven't found that would help.

And then this thing looks to us all very hard. As of now, so the task we're [INAUDIBLE] on is whole image classification. And so on that task it's possible that the kind of surrounding context-- so when you do a patch with an annotation, you kind of lose the context which it appears in. So it's possible that just by looking at the whole context every time, it's as good-- you don't get any benefit from kind of the zooming boxes. However, we're not evaluating on kind of an object detection type of evaluation metric. If you say how well we are catching the box. And if we were, we'd probably have much better luck with using the VGG annotation.

Because you might be able to tell some of those discriminations by like, this looks like a breast that's likely to develop cancer at all. And the ability of the model to do that is part of why we

can do risk modeling. Which is going to be the kind of the last bit of the talk.

Yes.

AUDIENCE: So do you do the object detection after you identify whether there's cancer or not?

ADAM YALA: So as of now we don't do object detection in part because we're framing the problem as triage. So there is quite a few tool kits out there to draw more boxes on the mammogram. But the insight is that if there's 1,000 things to look at, looking at 2,000 things you drew more boxes per image. And it isn't necessarily the problem we're trying to look at. There's quite a bit of effort there. And it's something we might look into later in the future. But it's not the focus of this work.

Yes.

AUDIENCE: So Connie was saying that the same pattern appearing in different parts of the breast can mean different things. But when you're looking at the entire image as once, I would worry intuitively about whether the convolutional architecture is going to be able to pick that up or whether-- because you were looking for a very small cancer on a very large image. And then you were looking for the significance of that very small cancer in different parts of the image or in different contexts of the image. And I'm just-- I mean, it's a pleasant surprise that this works.

ADAM YALA: So there is kind of like two pieces that can help explain that. So the first is that if you look at, like, the receptive fields of any given last receptive map at the very end of the network, each of those summarizes through these convolutions a fairly sizable part of the image. And so you are kind of, like, each pixel at the very end ends up being like something like a 50 by 50 image. That's by five total dimensions.

And so each part does summarize this local context decently well. And when you do maximum at the very end, and you get some not perfect but OK global summary, what is the context of this image? So something like, let's say, some of the lower dimensions can summarize, like, is this a dense breast or kind of some of the other pattern information that might tell you what kind of breast this is. Whereas any one of them can tell you this looks like a cancer given its local context.

So do you have some level summarization, both because of the channel-wise maxim of the end, and because each point through the many, many convolutions of different strides gives you some of that summary effect. OK, great. I'm going to jump forward.

So we've talked about how to make this learn. It's actually not that tricky if we just do it carefully and tune. Now I'll talk about how to use this model to actually deliver on this triage idea.

So some of my choices again, ImageNet initialization is going to make your life a happier time. Use bigger batch sizes. And architecture choice doesn't really matter if it's convolutional.

And the overall setup that we do through this work and across many other projects we're training independently per image. Now this is a harder task because you don't actually have the-- you're not taking any of the other view, you're not taking prior mammograms. But this is for kind of more harder reasons than that.

We're going to get the prediction for the whole exam by taking the maximum across the different images. So if I say this breast has cancer, the exam has cancer. So you should get it checked up.

And at each development epoch we're going to evaluate the ability of the model to do triage task, which I'll step into in a second. And we're going to kind of take the best model that can do triage.

So you're always kind of like, your true end metric is what you're measuring during training. And you're going to do model selection and kind of hyper patching based on that.

And the way we're going to do triage and our goal here is to mark as many people as healthy without missing a single cancer that we always would have caught. So intuitively kind of by taking all the cancers that the radiologist would have caught, what's the probability of cancer across these images, and just take the minimum of those and call that the threshold. That's exactly what we do.

And another detail that's quite relevant often is if you want these models to output a reasonable probability like this is the probability of cancer, and you train on a 50/50 sample the batches, by default your model thinks that the average incidence is 50%. So it's crazy confidence all the time.

So to calibrate that one really simple trick is you do something called Platt's Method where you basically just fit like a two-parameter sigmoid or just scale and a shift to just-- on the development sets to make it actually fit the distribution. That way the average probability you

would expect to actually fit the incidence. And you don't get this kind of like crazy off-kilter probabilities.

OK. So analysis. The objectives of what we would try to do here is kind of similar across all the projects. One, does this thing work? Two, does this thing work across all the people it's supposed to work for?

So we did a subgroup analysis. First we looked at the AUC in this model. So the ability to discriminate cancer is not. We did it across races. We have across MGH, age groups, and density categories. And finally, how does this relate to radiologist's assessments? And if we actually use this at test time on the test set, what would have happened? Kind of a simulation before a full clinical implementation.

So overall AUC here was 82 with some confident from 80 to 85. And we did our analysis by age. We found that the performance was pretty similar across every age group.

What's not shown here is the confidence intervals. So for example-- but the kind of key core takeaway here is that there was no noticeable gap in terms of by age group. We repeated this analysis by race, and we saw the same trend again. The performance kind of ranged generally around 82. And in places where the gap was bigger, the just confidence interval was bigger accordingly due to smaller sample sizes, cause MGH is 80% white.

We saw the exact same trend by density. The outlier here is very dense breasts. But there's only like 100 of those on test set. So this confidence actually goes from like, 60 to 90. So as far as we know for the other three categories, it is very much tied to confidence interval and very similar, once again, around 82.

OK. So we have a decent idea that this model seems at least with a publish of MGH actually serve the relevant populations that exist as far as we know so far. The next question is, how does the model assessment relate to the radiologist's assessment?

So to look at that we looked at on the test, if you look at the radiologist's true positives, false positives, true negatives, false negatives. Where do they fall within the model distribution of percentile risk? And if there is below the threshold, we've got to color it in this kind of cyan color. And if it's above the threshold, we're going to color it in this purple color. So this is kind of triage, not triage.

The first thing to notice-- this is the true positives-- is that there is like a pretty kind of steep drop-off. And so there is only one true positive fell below the threshold in a test set of 26,000 exams. So none of this difference was statistically significant. And the vast majority of them are kind of this top 10%.

But you kind of see, like, there's a clear trend here that they kind of get piled up towards the higher percentages. Whereas if you look at the false positive assessments, this trend is much weaker. So you still see that there is some correlation that there's going to more false positives the higher amounts, but much less stark. And this actually means that a lot of radiologist's false positives we actually place below the threshold.

And so because these assessments are completely concordant and we're not just modeling what the radiologist would have said, we get an anticipated benefit of actually reducing the false positives significantly because of the weight of disagreeing.

And finally, kind of aiding that further, if you look at the true negative assessments, there is not that much trending between where it falls within this. So it shows that they're kind of picking up on different things and they're-- where they disagree gives them both areas to improve and ancillary benefits because now we can reduce false positives.

This directly leads into assimilating the impact. So one of the things we did, we just said, OK. If people retrospective on the test set as a simulation before which truly plug it in, if people didn't rebuild the triage threshold-- so we can't catch any more cancer this way, but we can reduce false positives-- what would have happened?

So at the top we have the original performance. So this is looking at 100% of mammograms, sensitivity was 98.6 with specificity of 93. And in the simulation the sensitivity dropped not significantly to 90.1, but significantly improved to 93.7 while looking at 81% of the mammograms. So this is like promising preliminary data. But to reevaluate this and go forward, our next step-- let's see if-- oh. I'm going to get to that in a second.

Our next step is we need to do clinical implementation to really figure out-- because there's a core assumption here is that people read it the same way. But if you have this higher incidence, what does that mean? Can you focus more on the people that are more suspicious? And is the right way to do this just a single threshold to not read? Or have a double ended with the seniors cause they're much more likely to have cancer.

And so there is quite a bit of exploration here to say, given we have these tools that give us some probability of cancer, that's not perfect, but gives us something. How well can we do that to improve care today?

So as a quiz, can you tell which of these will be triaged? So this is no cherry-picking. I randomly picked four mammograms that were below and above the threshold. Can anyone guess which side-- left or right-- was triaged?

This is not graded, Chris, so you know.

AUDIENCE: Raise your hand for--

ADAM YALA: Oh yeah. Raise your hand for the left. OK. Raise your hand for right. Here we go. Well done. Well done.

OK. And then next step, as I said before, is we need to kind of push to the clinical implementation because that's where the rubber hits the road. We identify is there any biases we didn't detect? And we need to say, can we deliver this value?

So the next project is on assessing breast cancer risk. So this is the same mammogram I showed you earlier. It was diagnosed with breast cancer in 2014. It's actually my advisor, Regina's.

And you can see that in 2013 you see it's there. In 2012 it looks much less prominence. And five years ago, really looking at breast cancer risk.

So if you can tell from an image that is going to be healthy for a long time, you're really trying to model what's the likelihood of this breast developing cancer in the future.

Now modeling breast cancer risk, as Connie earlier said, is not a new problem. It's been a quite researched one in the community. And the more classical approach that we're gonna look at other kind of global health factors-- the person's age, their family history, whether or not they've had menopause, and kind of any other of these kind of facts we can sort of say are markers of their health to try to predict whether this person's at risk of developing breast cancer.

People have thought that the image contains something before. The way they've thought about this is through this kind of subjective breast density marker. And the improvements seen

across this are kind of marginal from 61 to 63.

And as before, the kind of sketch we're going to go through is dataset collection, modeling, and analysis. And dataset collection we followed a very similar template. We saw from consecutive mammograms from 2009 to 2012 we took outcomes from the EHR, once again, and the Partners Registry.

We didn't do exclusions based on race or anything of that sort, or implants. But we did exclude negatives for followup. So if someone didn't have cancer in three years, but disappeared from the system, we didn't count them as negatives that we have some certainty in both the modeling and the analysis. And as always, we split patients into train, dev, test.

The modeling is very similar. It's the same kind of templated lessons as from triage, except we experimented with a model that's only the image. And for the sake of analysis, a model that's the image model I just talked to you before concatenated with those traditional risk factors at the last layer and trained jointly. That make sense for everyone? So I'm going to call that ImageOnly an Image+RF or hybrid. OK. Cool?

Our kind of goals for the analysis. As before, we want to see does this model actually serve the whole population? Is it going to be discriminative across race, menopause status, the family history? And how does it relate to kind of classical portions of risk? And are we actually doing any better?

And so just diving directly into that, assuming there's no questions. Good.

Just to kind of remind you, this is the kind of the setting. One thing I forgot to mention-- that's why I had the slide here to remind me-- is that we excluded cancers from the first year from the test set. So there is truly a negative screening population. That way we kind of disentangle cancer detection from cancer risk. OK. Cool.

So Tyrer-Cuzick is the kind of prior state-of-the-art model. It's a model based out of the UK. Their developer is someone named Sir Cuzick, who was knighted for this work. It's very commonly used.

So that one had an AUC of 62. Our image-only model had an AUC about 68. And hybrid one had an AUC of 70.

So you know, what is this kind of AUC thing gives you when you look using a risk model. What

it gives you is the ability to build better high-risk and low-risk cohorts. So in terms of looking at high-risk cohorts, our best model place about 30% of all the cancers in the population in the top 10%, and 3% of all the cancers in the bottom 10% compared to 18 and 5 to the prior state of the art.

And so what this enables you to do, if you're going to say that this 10% should actually qualify for MRI, you can start fighting this problem of majority of people that get cancer don't have MRI, and the majority of people that get it don't need it. It's all about, is your risk model actually place the right people into the right buckets.

Now we saw that this trend of outperforming the prior state of the art held across races. And one of the things that was kind of astonishing was that though Tyrer-Cuzick performed on white women, which makes sense because it was developed only using white women in the UK. It was worse than random [INAUDIBLE] for African-American women.

And so this kind of emphasizes the importance of this kind of analysis to make sure that the kind of data that you have is reflective of the population that you're trying to serve and actually doing the analysis accordingly. So we saw that our model kind of held across races and as well across-- we see this trend from across pre-postmenopausal and with and without family history.

One thing we did in terms of a more granular comparison of performance, if we just look at kind of like the risk thirds for our model and the Tyrer-Cuzick model, what's the trend that you see or the cases where kind of like which one is right that's kind of ambiguous. And what I should show in these boxes is the cancer incidence prevalence in the population. So the darker the box, the higher the incidence.

And on the right-hand side are just random images from cases that fit within those boxes. Does that make sense for everyone? Great.

So a clear trend that you see is that, for example, if TCv8 calls you a high risk but we call it low, that is a lower incidence than if we called it medium and they call it low. So kind of like you kind of see this straight column-wise pattern showing that discrimination truly does follow the deep learning model and not the classical approach.

And by looking at the random images that were selected in case where we disagree, it supports the notion that it's not just that the column is just the most dense, crazy, dense

looking breast, that there's something more subtle it's picking up that's actually indicative of breast cancer risk.

Kind of a very similar analysis we looked at as if we look at just by a traditional breast density as labeled by the original radiologist on the development set or on the test set, we end up seeing the same trend where if someone is non-dense we call them high risk. They're much higher risk than someone that is dense than we call low risk.

And as before, the kind of real next step here to make this truly valuable and truly useful is actually implementing a clinically seamless prospectively and with more centers and kind of more population to see does this work and does it deliver the kind of benefits that we care about. And viewing what is the leverage of change once you know that someone is high risk? Perhaps MRI, perhaps more frequent screening. And so this is the kind of gap between having a useful technology on the paper side to an actual useful technology in real life.

So I am moving on schedule. So now I'm gonna talk about how to mess up. And it's actually quite interesting. There is like, so many ways. And I fall into them a few times myself, and it happens.

And kind of following the sketch, you can mess up in dataset collection. That's probably the most common by far. You can mess up in modeling, which I'm doing right now. And it's very sad. And you can mess up in analysis, which is really preventable.

So in dataset collection, enriched data sets are the kind of the most common thing you see in this space. You find in a public data set that's most likely going to be like 50-50 cancer, not cancer.

And oftentimes these datasets collect can have some sort of bias within the way it was collected. So it might be that you have negative cases from less centers than you have positive cases. Or they're collected from different years.

And actually, this is something we ran into earlier in our own work. Once upon a time, Connie and I were in Shanghai for the opening of a cancer center there. And at that time we had all the cancers from the MGH dataset, about 2,000. But the mammograms were still being collected annually from 2012-- from 2009.

So at that time, we only had, like, half of the negatives by year, but all of the cancers. And all of a sudden I had to-- you know, I came from the slightly more complicated model, as one

often does. I looked at several images at the same time. And my AUC went up to like, 95. And I had all this, like, bouncing off the wall.

And then in-- you know, I had some suspicion of like, wait a second. This is too high. This is too good.

And we completely realized that all these numbers were kind of a myth. But this level of-- kind of if you do these kind of case control things, you can oftentimes, unless you're very careful about the way it was constructed, you could easily run into these issues. And your testing set won't protect you from that.

And so having a clean dataset that truly follows the kind of spectrum we expect to use it in-- i.e., a natural distribution, collected through routine clinical care is important to say will it behave as we actually want it to be used.

In general, the only-- some of this you can think through in first principle. But it kind of stresses the importance of actually testing this prospectively in external validation to try to see does this work when I take away some of the biases in my dataset, and being really careful about that. The common approach of just controlling by age or by density is not enough when the model can catch really fine-grained signals.

How to mess up in modeling. So there's been adventures in this space as well. One of the things I've recently discovered is that the actual mammography machine device that the machine was captured on-- so you saw a bunch of mammograms probably from different machines-- has an unexpected impact on the model.

So the actual probability distribution-- the distribution of cancer probabilities by the model is not independent of the device. That's something we're going through now. We actually ran into this while working on clinical implementation is like this kind of conditional adversarial training set up to try to rectify this issue.

It's important. So this is much harder to catch based on first principle. But it's important to think through as you kind of really start demoing out your computations. This will kind of-- these issues pop up easily, and they're harder to avoid.

And lastly, and I think probably one that's probably the most important is messing up in analysis. So it's quite common in the previous section in this field-- yes.

AUDIENCE: With the adversarial up there, just to understand what you do, do you that discriminate or predict the machine? And then you train against that?

ADAM YALA: So my answer is going to be two parts. One, it doesn't work as well as I want it to yet. So really who knows?

But my best hunch in terms of what's been done before for other kind of work, specifically in radio signals, is they use a conditional adversarial. So you're free to discriminate at both the label and the image presentation. You have to try to predict out the device to try to take away the information that's not just contained within the label distribution.

And that's been shown to be very helpful for people trying to do [INAUDIBLE] detection based off on Wi-Fi-- or not Wi-Fi-- but like, radio waves. And the [INAUDIBLE] but also, it seems to be the most common approach I've seen in literature. So it's something that I'm going to try soon. I haven't implemented it. It was just GPU time and kind of waiting to queue up the experiment.

And the last part in terms of how to mess up is this kind of analysis. One thing that's common is people assume that's it kind of like synthetic experiments or the same thing as clinical implementation. Like, people do reader studies very often. And it's quite common to see that when you do reader studies that it doesn't actually-- like, you might find that computer detection does a huge difference in reader studies. And it's-- Connie actual showed it was harmful in real life.

And it's important to kind of like, do these real world experiments that we can say what is happening and just them the real benefit that I expected. And a hopefully less common nowadays mistake is that oftentimes people exclude all inconvenient cases. So there was a paper yesterday that just came out that the cancer detection used a kind of patched-up architecture which would read more closely into their details, they excluded all women with breasts that they considered too small by some threshold for like modeling convenience. But that might disproportionately affect specifically Asian women in that population. And so they didn't do a subgroup analysis for all the different races, so it's hard to know what is happening there.

If your population is mostly white, which it is at MGH, and is at a lot of the centers that these colleges have developed, are reporting the average that you see isn't enough to really validate that. And so you can have things like Tyrer-Cuzick model that are worse than random and

especially harmful for African-American women.

And so guarding against that is you can do a lot of that based on first principle. But some of these things you can only really find out by actively monitoring to say, is there any subpopulation that I didn't think about a priority that could be harmed?

And finally, so I talked about clinical deployments. We've actually done this a couple times. And I'm going to switch over to Connie real soon.

In general, what you want to do is you want to make it as easy as plausible and possible for the in-house IT team to use your tool. We've gone through this with-- not like-- I don't-- depends on how you count. It's like once for density and then like three times at the same time. But I spent, like, many hours sitting there.

And the broad way that we set it up so far is we just have a kind of docker as container to manage a web app that holds the model. This web app has kind of a backup processing toolkit. So the kind of steps that all of our deployments follow and I look under unified framework is the IT application would get some images out of the PAC system. It will send it over to application. We're going to convert to the PNG in the way that we expect, because we kind of encapsulate this functionality. Run for the models, send it back, and then write it back to the EHR.

One of the things I ran into was that they didn't actually know how to use things like HTTP because it's not actually normal within their infrastructure. And so being cognizant that some of these more, like, tech standard things like just HTTP requests and responses and stuff is less standard within the inside of their infrastructure and kind of looking up how to actually do these things in like C Sharp, or whatever language they have, has been really what's enabled us to end block these things and actually plug it in.

And that is it for my part. So I'm gonna hand it back-- oh, yes.

AUDIENCE: So you're writing stuff in the IT application in C Sharp to do API requests?

ADAM YALA: So they're writing it. I just meet them to tell them how to write it. But yes.

So like, in general, like, there's libraries. So like, the entire environment is in Windows. And Windows has a very poor support for lots of things you would expect it to have a good support for. So there was like, if you wanted to send HP requests for like a multipart form and just put

the images in that form, apparently that has bugs in it in like, Windows whatever version they use today.

And so that vanilla version didn't work. Windows for Docker also has bugs. And I had to set up this kind of locking function for them to like, automatically table locks inside the container. And it just doesn't work in Windows for Docker.

AUDIENCE: [INAUDIBLE] questions because he is short on time.

ADAM YALA: Yeah. So we can get to this at the end. I want to hand off to Connie. If you have any questions, grab me after.