EllenZhong

Sat, 11/27 1:03PM 1:17:10

SUMMARY KEYWORDS

neural network, representation, images, ctf, model, latent, heterogeneity, training, terms, people, reconstruction, decoder, architecture, data, guess, function, learn, voxel, particle, encoder

SPEAKERS

Benjamin Himes, Ellen Zhong

B Benjamin Himes 00:13

This is cryo2go, your place to get the full scoop on cryo em related topics. And each show we'll take a deep dive, guided by the authors themselves. I'm your host, Ben Himes. And I'm really excited about what's going on in open access and open science more generally. This pod is my effort to help bring out the context and some details that don't always survive the publication process. The most updated info, you can catch me over on Twitter at cryo2go, here's the show. All right, welcome to episode seven of cryoi2go, if you listened to the prior episodes, you may have noticed that both the introduction has changed. And that right now there's no music. And those two things are related. They're also related to the fact that it's been since March of this year that I've put out my most recent episode. And that all ties into having bit off a little more than I could chew in terms of the post processing. And I decided in the interest of being able to share more frequently, that I would simplify the pod just to basically be the conversations they are. And I hope that you all still enjoy it. So I've got a handful of episodes that are recorded at the very end of the summer in 2020, just a little over a year ago. And I'd like to get those sorted out of the vault, so to speak, because they're all still good and relevant today. And very interesting. So I'm excited today to share a talk that I had with Ellen Zhong from MIT, she's just wrapping up her PhD. And the thing that really drives her work, you may be familiar with cryoDrgn, is understanding how to map out continuous heterogeneity in the samples that we see Frozen on our grids. And now she's kind of moved on, I think, to looking forward to Okay, now, what do we do now that I've come up with this great way? In cryo dragon to map out continuous heterogeneity? How do we use that information? We talked a little bit about that speculating in the future. Kind of another cool thing that we touched on in the episode are the prospects of using Hartley transforms versus Fourier transforms when you're training neural nets, and how sort of a different representation of the same information can be useful in speeding up your training. And we also touch a little bit on her time at DE Shaw and what she did prior to pursuing your PhD, and sort of have a little bit of a riff back and forth on the pros and cons of academia versus industry. And, you know, sorting out what, what to do in science. So with that, I will just turn it over to our talk we had, like I said, back in August, and I hope you'll enjoy the episode. You have been very busy. You know, yes. Originally, way back when you went to I think the first two or three people emailed because you had just released this paper about cryo dragon on bioarxiv maybe a few months before I think it was in the fall, right?

E Ellen Zhong 03:14

Yes. Okay, was actually the publication or I guess preprint release story is a little bit complex, because the the first version of the manuscript was the machine learning paper. And then Joe Davis, my advisor, who's in the biology department was like, no one's gonna understand this. So we rewrote some aspects of it and like incorporatedthe relevant contacts for cryoEM, I posted that on archive. And then once machine learning paper came out, because archive is the traditional venue where these papers are published that we had to put the archive paper and overwrite that one. And then there was the Bio Archive preprint, which is more focused on applications of cryo dragon to real cryoEM data. So that one I think, was probably after you first reached out to me. Although I don't really remember the timeline.

B Benjamin Himes 04:08

That might have been I think, February, I was just browsing around to refresh my memory. So

E Ellen Zhong 04:14

yeah, the whole Yeah, that it's like tricky to navigate the two domains between machine learning and Crimea, and figure out like, oh, which audience are we talking to today?

B Benjamin Himes 04:26

Well, that will be actually a good point for today, because even the bioarxiv paper has some meaty material. And at that, you know, I have a feeling a fair amount of crowd members are gonna go well, I've read that word. I, I've heard variational autoencoder before. But you know, if you want to cover a little bit of background while we're talking, yeah, ready to do that? Yeah. So real quick. So you mentioned Joey Davis there and then you also so you mentioned he's on the biological end of things. So you're also supervised by Bonnie Berger.

E Ellen Zhong 04:59

Yes. I'm a PhD student in the computational biology program at MIT, which is this very small interdisciplinary program, like my year has six people in our cohort. And people are kind of scattered all throughout either computer science, or biology or biological engineering. So I'm co-advised between Joey Davis and Bonnie Berger. The former is a professor of biology department. And Bonnie is a professor at MIT computer science department. Gotcha. Yep. So and then like, I work on machine learning algorithms for cryoEM. And so for me, it's like a really great place to have both biological or just even a lab who, whose people will talk about, like, experimental biology in these in autophagy, in particular, and then my lab in CSAIL. They, it's nice to be able to talk to people about the computational aspects and algorithms and software and things like that. That's kind of like I have a foot in both worlds, which is the story of this whole project, I would say,

Benjamin Himes 06:07

that's great. And so I know, one thing that can happen in this sort of situations is can be easy to sort of have a foot in both worlds and then get lost in between them. Have you ever worried about that? Or do you feel pretty well routed in both places?

Ellen Zhong 06:22

Um, I definitely feel kind of like an outsider in both places. But I think it's one of the, it's, it's definitely an advantage and a disadvantage. It's nice to be able to put on whichever hat that I'm interested in. Yeah, whichever hat I'm interested in for that particular day, but it's also like, Okay, I have to be up to date in the literature of both the machine learning side and methods development, and the cryoEM community, and the cryoEM methods development, and then there's just like, computational biology, overall. Right. So it definitely lends itself to being like, Okay, who like, what, what am I? Like, which? Where do I even belong? Or what literature do I like, look into today? It's also really nice, because you can take ideas from one place and apply them to the other, and vice versa.

Benjamin Himes 07:21

Yeah, I mean, that sounds pretty relatable. That's actually that reminds me of one of the other reasons for starting the pod was, you know, I'm in a similar bit of a camp, you know, I had a little bit maybe more formal time spent in a biological lab when I was an undergrad, but then everything since then has been, you know, been behind the computer. And so I feel really out of place. So we start talking about, you know, like, I remember what a ribosome is, but you start asking me about cofactors, or how things actually work is pretty fuzzy. So getting down positions has been really useful for me personally.

Ellen Zhong 07:53

And every field has a sub field, and every sub field has its own jargon, how they call things. So you remember one of the first times I gave a talk about cryoDrgn, I think at Harvard's cryoEM club. People asked me about symmetry, but they used all of these terms that I'd never heard of before, like this. Like it's symmetric, but it's pseudo symmetric. And I'm just like, right, so what does that actually mean? Right.

Benjamin Himes 08:20

Sounds like a pseudo term.

Ellen Zhong 08:24

So you you also work on methods development. I haven't looked into emClarity, but I, I've heard it's for tomography.

Benjamin Himes 08:32

Yeah. So that's looking at subtomogram averaging. And that was sort of the end result of my PhD work, which I did. It was a joint program at Pitt and Carnegie Mellon. And so yeah, it was sort of similar in both worlds, but I was, I was I only had one supervisor, which made it I guess, a little bit easier, because I then also wasn't totally split between camps. But, yeah, I mean, I try and support that where I can, but, you know, supporting something like that, that a lot of people are trying to use, which feels great. But at the same time, I also learned a program while I was writing that, so it's a bit of a mess, if I'm being honest. Right. And now there's some better solutions. I hope. Maybe I shouldn't say this on on air, but I kind of hope people migrate away from it. I think we can edit this out. Yeah, exactly. Right. So yeah, but I've been working now with Niko Grigorieff for about three years and doing development inside system, which is it's a pretty heavy single particle package, but it's branching much more into being a sort of generic cryoEM toolbox. So, yeah,

Ellen Zhong 09:45

I definitely have to say it was really hard. I think when I first joined the labs, I spent a couple of months just reading up on all the literature, as one does, but then in terms of criteria methods, there's just so many and they all it's Very confusing how they're all related to one another. When you start out, it's like, okay, right and reconstruction, what is the alignment? Are they actually different?

Benjamin Himes 10:09

Well, and this is one of the things actually that I popped in my mind when I was reading through your Bio Archive paper is you're talking about branch and brown are oh we'll edit that out, uh, branch and bound. And the first thing that popped in my head was like, Oh, I read about that in the cryo spark paper when that very first came out. But then I stepped back and thought about it. I was like, but that isn't so different from what we do in sort of our own, maybe ad hoc, hand wavy way and a lot of the traditional cryoEM processing. Same thing with the frequency marching. Yes. Which I think it's important to have the right names to call things so that you can communicate, particularly with people coming into the field. So it's, that was one of the nice things, reading through papers, assigning a name to this, these things that, you know, we do. But yeah,

Ellen Zhong 10:52

and as a sneak peek, what something that I've been working on now is kind of removing all the branch and bound code from the Abinitio reconstruction code paths, because there's just certain ways that it doesn't really make sense for the whole neural network model. Okay. And doing the more I guess, heuristic based, let's just try a bunch of poses can be the error, see which one, just like take the top N. For the next iteration, it actually is just much simpler in terms of implementation and way faster. And that's like more of the bottleneck when it comes to Abinitio heterogeneous reconstruction for cryoDrgn,

B Benjamin Himes 11:33

that's really interesting. So that's actually one thing. You know, I know there's an avenue and rely on. But I think, as far as I know, the two sort of most commonly used are cryo Sparc. And what Tim, Tim Grant has implemented in cisTEM. And what he did was sort of this, you know, random exploration, almost like a stochastic thing, where it's just sort of getting out where you're going to there. So the sneak peek is cool, why don't we take a step back, then, and then I want to give sort of a big picture view of what is cryoDrgn. And then we can sort out some details as we go through it.

E Ellen Zhong 12:06

So cryoDrgn is a neural network based heterogeneous reconstruction method for cryoEM. And there's kind of two aspects of it that I like to describe. The first is the neural network representation of structure, which is a bit technical, and the Machine Learning Community is a lot more interested in that modeling aspect. And then there's the auto encoder, variational auto encoder aspect of the method, which is how we learn the heterogeneous landscape. And so I think that part is the more new aspect for the cryoEM community. So a VAE or even just an autoencoder, the way I like to think about it is that we have two neural networks, and it's just doing a nonlinear dimensionality reduction on our data. And so in the context of cryoEM, there's an cryoEM reconstruction, there's a little bit more detail, because we have two dimensional images. And at the end of the day, we want three dimensional volumes. But even if we just ignore that for a second and think about how the VAE, the dimensionality reduction, we just use a auto encoder to encode the particle images into some latent space. So this is just a low dimensional abstract representation of the heterogeneity. And then we have a decoder neural network that takes this latent space representation and reconstructs the input image or three dimensional volume.

B Benjamin Himes 13:34

Let me get a pause there, and there's two things to break down. So you said a nonlinear dimensionality reduction. So to tie that into something that, you know, maybe old school cryoEM'ers might recognize is that something like PCA would be an example of a linear dimensionality reduction? Okay, so it's sort of a nonlinear advance from PCA. Okay, and then could you just also define what a latent space is?

E Ellen Zhong 14:02

Yeah. So a latent space, I guess comes from the whole latent variable notion in machine learning. So that is like a jargon term taken more from the machine learning side of things. But the main idea is that we have some complex observed distribution of data, like our three dimensional volumes, or maybe just a simplified example, just multimodal distribution, but this can be explained in terms of more simple degrees of freedom. So you have like it specifically in terms of a latent variable model. So you have some latent or unobserved I guess, Oracle, doing things to your data. And then from that latent, I guess, latent description of the generating process, you have your observed distribution, which is complex. I don't know if that is a more straightforward way of answering that. Question.

Benjamin Himes 15:00

No, I think that sounds pretty good. And then so for our physics friends, the Oracle, that's like Maxwell's demon, do you just some something that's back there flicking atoms around that you don't really have to know how it works? Or but you want to try and understand the result of its flick? Or is that not true?

Ellen Zhong 15:16

I guess I'm not super familiar with Max, Maxwell's demon. Okay, so maybe the idea, right, I guess like, one way, I think the most easy way to think about it, if we come from a structural biologist perspective is the whole conformational energy landscape idea. Like we have some low dimensional, like, energy landscape that are complicated 3d structures are generated from and so if we have, so this, there's like, there is a mapping that relates this like simple surface to the three dimensional structure. So can we learn that mapping? Okay, that's one of the goals of cryoDrgn.

Benjamin Himes 15:55

Great, I think that's a really nice way to put it.

Ellen Zhong 16:00

There's, it's very interesting, there's just like a million different kinds of framings for this problem. And I've definitely experimented with like, one or the other, depending on the audience, and just depending on the exact mood of the moment. So yes, so like cryoDrgn, there's the neural network aspect of cryoDrgn in terms of how we represent 3d structure, which is one aspect of the method. And then the other aspect of cryoDrgn is this variational auto encoder architecture, which has this latent representation as the intermediate bottleneck layer. And so what the VAE does, one way of thinking about it is to learn this low dimensional representation or data. And so that was this latent space that I mentioned. And in many ways, it's similar to a lot of the other approaches for heterogeneous reconstruction, where you're trying to learn the ensembles. There's like manifold em, there's 3d VA, which is a PCA based approach. And so the key here is that our latent representation of the data set is mapped back out to the three dimensional space via a neural network. And so I think that's one of the key advantages of cryoDrgn is that using a neural network, which, in theory has very few limitations on what kind of functions it can model. In practice, that's obviously something to explore and something to optimize the architecture. But anyway, we can model very complex nonlinear functions, including conformational changes, presence and absence of particular sub units. And increasingly, as we've been applying the tool on more and more real datasets, we definitely see a lot of just like empirical evidence that this approach is very nice. And just like magically works, which is, which is very similar to many other deep learning applications.

Benjamin Himes 18:05

Right. So now you said you one of the nice things about the neural net is that you can fit, you

Right. So now you said you one of the nice things about the neural net is that you can fit, you know, basically any function that has any level of complexity. So by you know, and you're also using a fully connected network, right? So is there did you run into trouble giving it that many degrees of freedom with either lack of convergence or overfitting?

Ellen Zhong 18:29

Yeah, there are definitely cases like or corner cases, or just regimes where it doesn't work super well. There's the architecture itself. So like, if you don't have a large enough of a neural network, then you're just parameter limited, and you can't model like very high resolution structures. So that was one thing that we explored in our paper is okay, if we increase the architecture of the neural network that we can learn higher resolution details. One other limitation that's been that, I guess, was also in the paper, although there's just not enough space to talk about like, all of the aspects is how large of a block you can model. So I think we tested this on the Plasmodium falciparum 80S resume, which full resolution images are a 360 by 360, that were deposited on EMPIAR. And if you think about the number of voxels in the 360 cubed, volume, let me just do that real quick. Which is about 46 million voxels. And then I think the largest neural network architectures we were testing is about an 11 million parameters. So there's kind of a mismatch there in terms of the complexity of the function that we're using to model this very large volume. There's some kind of inherent limitation there that we found empirically. And then one, I guess, early version, very early version of the method was Just purely testing a fully connected neural network, the actual method that we published is what we call in the machine learning community now, or I guess, what has now been called a positional encoded MLP. So MLP stands for multi layer perceptron, which is just just another jargon word for fully connected neural net. Anyway, we use a position positional encoding function as the input layer to our neural network. And that gives this as much richer features ation of 3d Cartesian coordinates, so that the neural network can more effectively learn this function. So

Benjamin Himes 20:38

that was Yeah, I was just gonna ask Is that so instead of learning, this 2d image might come from this set of image intensities, you're saying this 2d image represents these Euler angles and shifts? Or what does that exactly mean? The coordinate coordinate?

Ellen Zhong 20:55

Oh, the, so this is kind of even removed from the 2d images, but just thinking about how to use a neural network to model 3d volume, okay, the input layer contains the 3d Cartesian coordinates that specify like, Okay, where are you in this volume. And so in the voxel array that's kind of given non parametrically, just via, like the index into the voxel array, but our neural network learns this, like function from 3d Cartesian coordinates. So we input xyz coordinates, and it returns electron density at that, or what I've learned from cryoEM community density, that location, okay, so right, so what I was alluding to is that if you just naively input the Cartesian coordinate, then you can learn this function, but it's much harder to learn than if we featurize, those coordinates using this positional encoding function, which is really admittedly like a minor footnote in the paper. But is definitely one of the key architectural details that make it work.

B Benjamin Himes 22:06

And so you just touched on another one of the, I think, real nice opportunities to get to talk about this. I mean, obviously get to give talks and travel around, but having something nice and recorded. It's almost like an addendum to the method section, where you get to fill in these little details that, you know, well, this was actually super important. And I spent, you know, two months on it. But

E Ellen Zhong 22:24

yeah, yeah, I Yeah, definitely. And also, a lot of the talks that are given have such like, onerous time limits. And if you're introducing the method to the cryoEM community, like no one's gonna care about like the, like minor architectural choices in your decoder network. So it is, this is like a great space to talk about some of the details.

B Benjamin Himes 22:50

Okay, so with the, the positional encoder did I get that, right?.

E Ellen Zhong 22:58

So, positionally positional encoding, yes, yeah.

B Benjamin Himes 23:01

So unpack that just a little bit for me.

E Ellen Zhong 23:05

Yeah, so I guess just to clarify, there's the encoder/decoder architecture, but then I'm like, kind of overusing the word encoding. But the input to the decoder is another positional encoding of Cartesian coordinates. This might be easier if I drew this out, which is maybe one of the downsides of the podcast format. But the there's so there's like a encoder neural network that takes our particle images and maps out to the latent space. And then our decoder, NLP or neuro that has takes the latent variable from the encoder, and then the positionally encoded Cartesian coordinates. So I think maybe I can just say it takes the Cartesian coordinates, because there's just like a fixed mapping.

B Benjamin Himes 23:59

And this is all, maybe figure two. So I just open that up, so I could picture it a little easier. I think maybe it's mapping out how this no work actually looks.

Ellen Zhong 24:08

Yes. Figure two in the archive paper. I think figure one is just like single particle reconstruction background. Yes, so figure two of the machine learning paper has the architecture. And you can see that I'm also drawing the image like how it's trained. So instead of reconstructing a volume, where we're just reconstructing a slice in free space, so this is taking the Fourier slice theorem into account. And then you can see that there's like my, like PowerPoint, art of a lattice, and then a little circle that maps with the peace sign kind of that maps it to this like larger vector, but also gets input into the decoder and so the positional encoding featurization, it takes a Cartesian coordinate, which is just three numbers, a vector of three numbers. And then maps features that using a basis of sinusoids. So now instead of just three coordinates, now we have, I don't know, it kind of depends on the box size that we're modeling. But let's say we're modeling a 128 box size. Now it's 128 times three, were the first, I guess, like the first element of a positional encoding is the sine of that value. The second value of the positional encoding is the sine of two times that value. So now it's like a sine wave that has twice the frequency.

Benjamin Himes 25:56

And there's this because you're representing things in Fourier space. Is that sort of what inspired this? It's the chunk of a discrete transform, or is it more general than that?

Ellen Zhong 26:06

Yeah, it's kind of like, it's not because the data is represented in Fourier space. And it's definitely inspired by that. So but like maybe another way of thinking about it is instead of giving it just one, instead of getting the neural network, just one value to map to the actual value of the function. Now, we're telling it kind of more of the location information of that value, because the first element of positional encoding function tells it like, Oh, are you more on like the left side of the box or the right side of the box, you just think about what a sine wave looks like. And then the next higher frequency element will be will give you slightly more information about like, Oh, are we in the left side of the left quadrant or the right side of the left quadrant. And so this Fourier featurization gives it more information on like, a lot more data, it's definitely not intuitive to think about as a human. But if you just bought a neural network, now we're just giving it like so many more different representations of like x equals one, then just one number. And so I think that really helps with neural network learning. And so like, why this is now called a positional encoded MLP is actually more from the machine learning community, which has taken this architecture and like applied it on all have a lot of real, I guess, like natural image and natural volume modeling. And so there's actually been a couple of follow up papers now from more than traditional machine learning or computer vision community, where they explore this particular kind of architecture, and evaluate why it is that this free feature zation really helps with neural network training.

Benjamin Himes 27:53

Alright, so you've got the positional encoding, and that's expanding into this sinusoidal basis. And that feeds into the decoder network. So real quick, he said, you know, choosing the size of the network is really important. So you had a big enough representation, enough features to represent the space you're trying to learn about. I think in the bioarxiv anyway, it says you have 10 layers, is that a big network? I'm not familiar enough with neural nets to really know.

Ellen Zhong 28:23

Ah, so the write the arxiv paper, I believe we use like 128 Sorry, 10 layers, and each layer is 128 nodes per layer, which is like a very silly architecture, no one use that no one who's listening to this do that. And I think those are just random numbers that I picked literally on day one, when I first started this and it just like, the results are fine. And in the machine learning community. For the machine learning paper, we weren't really interested in like pushing the resolution limit or anything like that. Just seeing if it actually just seeing if it works. So in the later follow up work, applying cryoDrgn real data sets, there is the default architecture, which is three layers of 256 nodes per hidden layer. And that one is such the default because it's relatively small. And so it trains really fast, but it seems to have good empirical performance on smaller images. So like 120 by 120 images. And then a larger neural network that we've been trying would be more like three hidden layers of 1024 nodes per layer. And so I think I have a comparison of the number of trainable parameters and each of these networks but the 1028 by three I think provides a good trade off between Okay, we have more representation capacity here, but it's an it's slower to trade but not like too slow. We could also increase the dimensions of this neural network even further, but then it becomes very slow to train. And that becomes the bottleneck, I think of actually discovering interesting things.

Benjamin Himes 30:12

So how long so if you're training this on every data set, how long, you know, with, I don't even know what typical hardware is kind of spoiled with computers. So whatever typical hardware means, maybe it's a couple GPUs. How efficient is the training at this point?

Ellen Zhong 30:29

I would say the training is very efficient for my purposes, and I'm also very spoiled. One thing that I've learned from users is, there's a huge difference in whether you can load all of your data set into memory versus not. So if you can load all your data set in memory, then training, like that's kind of the regime that I've been looking at. The training is like, proceeds at a very reasonable clip, like for the smaller architecture, smaller architecture trading runs, it'll take on the order of a few hours for training a model. This also scales linearly with the size of the size of the data set. So I'm also looking at data sets that that kind of talk about 500k particles. So it's a few hours, but then the larger models would be maybe like overnight, and I'm definitely more limited in terms of like my own analysis and all the things that like, yeah, all the it's definitely human limited in my case versus trading time limited. Sure. And that's on one GPU. So one thing that I've added recently to the codebase, is GPU parallelization. And I actually added this few months ago, but I just haven't advertised it. Because I, because there are differences that there are like training differences that happen when you train over multiple GPUs. And I just like haven't had the time to evaluate that in detail.

Benjamin Himes 32:02

Right? And you mean, you're splitting the you're actually splitting the model over multiple GPUs? Right?

Ellen Zhong 32:08

Yeah, so we're splitting them, the model is replicated across the multiple GPUs. And then the data. So the training data is split for a particular for like a batch of images. Now that's split into onto their four GPUs, for instance.

Benjamin Himes 32:25

I see. So the the model itself, the full model still has to fit on each GPU. But I guess in this case, we're talking about three layers and, you know, 1024, so that shouldn't be a problem.

Ellen Zhong 32:35

Right? Okay. But you can, so for the neural network case, it can change the training dynamics, because now, instead of a mini batch of eight images now, or now we have like a mini batch of 32 images, so you go four times faster, but then you have four times fewer model updates. And so maybe, that, like absolute wall clock speed, or speed up for like trading a particular model actually leads to slightly lower resolution models, then you need to compensate by training for longer. And so that is something that I that relationship I haven't fully tested.

Benjamin Himes 33:18

So the reason that the batch size itself can affect the outcome. So if you're looking at x images versus 3x, there was actually it might not be the same.

Ellen Zhong 33:29

Yeah, exactly. So this is right, this is like one of the benefits, or one of the features of training deep neural networks is your mini batch size, can make a difference in how the training dynamics of the model, just like another one of the hyper parameters, and the original default that it's set to was just based on some back of the envelope calculations in terms of like, how to optimize the number of model updates per second, so that we're not like bottlenecked, in terms of like reading the data versus updating the model. And we need like, or like GPU memory. So there was like some amount of just back of the envelope calculations that went that went into deciding on a default number, but I just haven't done that for multi-GPU training.

Benjamin Himes 34:16

Yeah, that's actually really good to know. So I tried, you know, one of the things that I think would be really cool is if you could simulate data well enough, maybe you could just have a network that learned protein structure if you trained it on every protein that we know. And so, you know, I started playing around with a simple test case, but one of the things I didn't put into my random hyper parameter sweep was the batch size, which I guess just shows how naively I'm just going about things but that's really good to know. So I've already learned something very useful.

Ellen Zhong 34:47

Let's see we were talking about decoder positional encoding the training dynamics, I like went off in one direction because he started talking about GPU parallelization and then I love this format because I can like kind of go in depth into like one little thing.

Benjamin Himes 35:00

So actually, before we leave that, since we can display things together, this is still implemented in pytorch. Right? Yes. Okay. And I think in the version that I was looking at, I'm sorry, I didn't read the updated paper. So I'm feeling a little underprepared. But

Ellen Zhong 35:18

I will send it to you, that would be great, actually. So there's like an updated the bioarxiv, which we posted in March. And then I've been working the last like month or two on revisions for that manuscript. And so there's like a new version that I'm about to post or update on my archive, if the journal lets us that I think is way more comprehensive, and I love the figures so much more. So very excited about this new version, I can send that one to you,

Benjamin Himes 35:46

that would be great. So I need to figure out how to set up. Even if it's specifically for articles, I've already downloaded some sort of RSS that updates me and says, Hey, this has changed, because I tend to just go back to Mendeley, and, you know, I'll be looking at the original version that came out.

Ellen Zhong 36:03

Yeah, so weird. Like, the whole preprint thing, like adds this level of complexity to the paper / publication process. Strange. Like, has advantages too, but it's also like, there are also very large disadvantages that just like yeah, I guess it's really a world that we live in

Benjamin Himes 36:24

in terms of time management, and yeah. Okay, so I was gonna ask about the the pytorch. So I think in here, it says, a user friendly version is in preparation or something like that. So is the code. Is that on GitHub or somewhere? And when you say user friendly, Does that just mean like it's working? Or are you working on something I don't know, what is user friendly?

E

Ellen Zhong 36:47

So the code I released a few months ago, like also beginning of summer, actually, the code I released in March when we posted our biorxiv version of this paper. And so now it's like, freely available, anyone can download it, use it, the source code is all open. And I would like to thank that user friendly enough. I made it there's like, obviously, beta users, a lot of people reached out after we posted the first paper. So I had like good feedback on like, what inputs were just really confusing to people and what what I should like, figure out some different like format, executable format form. And so I think the, the version that's now on GitHub should be relatively easy to use, I would say, with like, a couple of aspects of who knows. If you have familiarity with like Anaconda environments, it should be pretty simple to install just conda install pytorch. Hopefully, the GPU or CUDA versions and your GPU box are compatible with like the thing that's first downloaded. And then it should be pretty simple just to install cryoDrgn and get it running. There's just a few like pre processing steps that are provided with like executables to parse out the appropriate CTF information from Star files or from cryoSparc CS files. And then everything is kind of boiled down into one training script. So cryoDrgn trainVAE, and then a million settings. But I think I've tried to make all of those have reasonable defaults. And then, in terms of analysis, there's like all sorts of interesting things you can do. And I tried to kind of package up some of the standard things that I do just to generate structures and analyze the latent space. But that's obviously very data set dependent. So I tried to keep it very, like modular and flexible. One of the things that's provided in the analysis pipeline for cryoDrgn is just copying over a template Jupyter Notebook. So this is just like a Python notebook that you can use to interactively explore the latent representation. And it like somewhat presupposes, you know how Python works and Jupyter notebooks but hopefully, everything is laid out in a very kind of understandable way. With like little widgets that let you kind of click through different representation or different plotting plotting visualizations.

B

Benjamin Himes 39:32

Awesome. Well, I'm going off to put that to the test and then we'll have feedback before the the actual final episode goes on there. See where I got

E

Ellen Zhong 39:41

Yeah, that would be great. I love this

B

Benjamin Himes 39:47

Yeah, I think going from something that makes sense to you and then something that it's it's one of the things with him emClarity anyway, that it amazes me how cool people think about problems like they think about things in a very cool way, but also in ways that totally break my thought process. And that then translates into breaking my code, which that's probably partially just, you know, not really, still even fully knowing what I'm doing in terms of creating something that's a robust software solution versus something. That's a cool algorithm.

Ellen Zhong 40:17

So yeah, I always tend to just re-implement everything myself instead of using other people's code to avoid this problem, which is obviously extremely inefficient. But so far, so good,

Benjamin Himes 40:30

right. And at some point, it becomes more efficient after you get enough of your own libraries built up. Yeah. Okay, so let's, let's pull it back into the discussion of the model. So we had gotten through from the positional encoding into the decoder. And then we still need to talk about how we go from the decoder to the actual reconstructed volume.

Ellen Zhong 40:53

Yeah. Yeah. So the decoder right is this neural network that takes our latent variable and three Cartesian coordinates. So to generate your voxel array that you can then load in chimera, you would just evaluate, like a 3d array of the different Cartesian coordinate positions at a fixed to like a constant value of our latent variable. So what this is doing is kind of like evaluating our function at this location in latent space. But like rasterizing across a 3d array of

Benjamin Himes 41:32

so voxel values, so is the decoder, what's actually translating from the three coordinates and applying this layer of the sinusoidal function? And that's what gets put in, or is that not quite right?

Ellen Zhong 41:44

The input to the decoder is this like these featurized coordinates. Um, but it's fair like that is fixed. It's just like a fixed function that just converts it. So I don't really...like the way I talk about it is that like, the input to the decoder is the 3d Cartesian coordinates, just because that first, like, featurization is just constant? Like,

Benjamin Himes 42:06

I see. And that was the peace sign or the yin yang thing. And the version I'm looking at is the features. I got it. Okay. Okay. Okay. Honestly, it makes perfect sense. So we got this, um, this representation? And then yes, you just rasterize you build up your model. And then since you're learning this for every dataset, this is iterative, right?

Ellen Zhong 42:29

Yeah. So you'll train like a completely new model from scratch for a new data set, right, or even

Yeah. So you'll train like a completely new model from scratch for a new data set, right, or even for like the same data set with like a different pixel size. Like, if you wanted to now move from your lower resolution images to higher resolution images, you would still need to train a new model, because the neural network has learned, like, yeah, the neural network we've already trained is for that lower resolution model. So one way to think about it is that the neural network is the result of the reconstruction. So like, in the traditional approaches, you have your voxel array or voxel res at the end. Whereas now after you do a cryoDrgn reconstruction, the output is the 3d voxel, or sorry, the weights of the neural network, which can then produce the 3d voxel array.

B

Benjamin Himes 43:18

That's actually an interesting way to think about it. And that makes a lot of sense. So are you able, so thinking about it from that context, where the real result is actually the set of weights that define the network? Although that networks been trained for a particular molecule and a particular pixel size and all the, you know, the particulars of a data set? Can you use those weights in any kind of transfer learning? Or is that are they to, like, uniquely defined to actually be useful?

E

Ellen Zhong 43:45

Um, I haven't explored that in that much detail, because there's a lot of, yeah, there's just so many ways it can go wrong. So I haven't really explored that. One thing that Joey is excited about is Oh, can we take our encoder neural network? So we haven't really talked about encoder as much but can we encode. So the encoder like learns this mapping from particle images to the latent space? And so in many ways, it's doing like the dimensionality reduction and giving you like a lower dimensional way of visualizing or just interpreting your dataset. And he's like, Oh, can we take that and apply it on other datasets or on other imaging sessions. I'm also not super optimistic just because the encoders learning very specific features of this particular data set it was trained on and there's just so many variations between imaging sessions, like the absolute grayscale images may be different. And if you just think about what like the neural network, as much as we like to think of it as an Oracle is just like a set of linear like, transformations plus nonlinear like RELU, non linearities, and like, if you just give it different values of those images, I can't imagine that it was no. Like, if it's not been trained on that, that I can't imagine it would know. Like anything like that it would do anything meaningful. But you never know.

B

Benjamin Himes 45:16

Right? So actually makes me think of, you know, one of the things I ran into is that I assumed, in my naive experiments that if I took care of the at least the CTF multiplication, so the phase flipping at the outset, that'd be one thing the neural net will have to learn. And by taking off that dimension, it would make things work better. But it turned out at least in my case, it did just as well, if not maybe a little better. If I didn't, I didn't even ask it to specifically give me any information about the defocus. It just ignored it. So in your paper, I think you said sometimes you do sometimes you didn't. Was there? What did you find with pre or no application of the CTF at a time?

E Ellen Zhong 45:58

Yeah, maybe that was because I used a lot of synthetic datasets in the initial paper, which maybe I don't remember if I even applied CTF to the initial data sets. So that may have been like that language that I was writing about. The other thing that we do do is for the real datasets, I always apply a CTF at the end to compare it to the input image, but for the encoder. So if you think about what the encoder has to do, it has to learn this mapping. But even for a particular particle or particular conformation, if we have particle images from different views, if it's in the same conformation, then ideally, it should be mapped the same location in lane space. And so the job of the encoder is actually a pretty complex function that it has to learn it has to learn a pose invariant representation of the heterogeneity, the structural heterogeneity. And so if you throw in things like the different views that pose heterogeneity, and like also, CTF heterogeneity becomes harder and harder to learn this encoding function. And so what we do is I actually phase flip the input images before feeding them into the encoder. And seeing if that and I think, I don't think I ever did like a direct evaluation of like, whiffs baseball thing or without facelifting just kind of assumed that like, Okay, if we didn't face flip, that would add, like all sorts of, like, weird signal into the images that might be hard to unlearn, you know?

B Benjamin Himes 47:38

Well, no, I mean, that it makes perfect sense. That was my thought process too, one thing, and I know, you're super busy, so you probably not have time for extra experiments. But if you were to revisit that idea, one thing you might try is, instead of just phaseflipping, to do just even like an ad hoc, kind of Wiener filter, so it wouldn't work right at the CTF zeroes. But even if you do the face flipping, you still are going to have these regions where there's obviously dampening going on as you get closer and closer to a CTF zero. And I mean, the trade off is that you can amplify noise, but you also maybe, air quotes here that you can't see, erase some more of the some more of the disturbance from the CTF. Yeah, yeah. So it'd be definitely, it'd be interesting if it impacted the encoder or not.

E Ellen Zhong 48:26

Yeah, and there's also a lot of like, follow up tweaks that I probably should make to the like noise model. Like right now I'm using a white noise model. And then like, there's no like variation, uh there's no changes in the variance based on like the CTF, which is something that is done in like traditional methods. And so, there's like lots of maybe just like low hanging fruit in terms of the actual modeling part that I can probably implement, although that would maybe require me understanding those a little bit better.

B Benjamin Himes 49:04

So did you find if you're modeling things, just with, you know, the straightforward additive white Gaussian noise? Did you have any problem generalizing from the simulated images to real data, and that's where some of these tweaks came from, or the tweaks just kind of come from a little more, I guess, experience and testing on more data sets.

E Ellen Zhong 49:23

I think that the tweaks came Yeah, mostly empirically, just like testing on initial datasets. The initial look like it's surprising to me actually, like just the simple the simplest, additive Gaussian white noise model works for the real data. And so far, like on a lot of on a number of real data sets, it seems like there hasn't been too like. I haven't noticed any weirdness because of our noise model. Like perhaps if I used a more expressive noise model. We could learn faster or learn like better. Yeah, I think we would just learn faster if I had like maybe a more expressive most model. But, um, that does add additional complexity. So I haven't really explored that in detail.

B Benjamin Himes 50:11

Okay. Let me think there, actually. So one more question about the, the effect of the input images on the encoder then, or even just the final result? So, have you noticed any correlation with how well centered the particles were in picking? So in other words, does it learn angles with equal ease to positional offsets? Or did you notice any kind of difference there, like big offsets?

E Ellen Zhong 50:36

Yeah, I haven't noticed, like specifically that that would be an issue, but it is definitely on my mind. So like, if your particles are not well centered, then the encoder also has to learn that like, Okay, your particle could be like all the way over here. So that's definitely something to consider. The other, right, so like, the software that's actually released is the version of the code that has poses already assigned from a consensus reconstruction. So it's assuming that you've already like you can get a like, believable consensus reconstruction from your protocols. And then it just already has the poses and already has the positional shifts. And so what I do now, what I do in that case, is I just sent her all the images, the idea being also just to remove that degree of freedom from the learning process.

B Benjamin Himes 51:29

That makes a lot of sense. Okay, cool.

E Ellen Zhong 51:33

Yeah, there's like all sorts of additional challenges in the Abinitio case. Like all these, like tiny little things like this, where like, Oh, should we like the recentering them on the fly each time? Or does that add more complexity because now the encoder is like in the beginning, learning this, like pose, like varying function, and then it's like changing what it's needing to learn or not. So I'm definitely still in the, I'm still in the process of figuring out like, all those details.

B Benjamin Himes 52:06

So if you just saw basically, you can go through and use I guess, cryo Spark, or you also parse rely on store files. And then, so as long as you get an image stack, you expect them all to be cut out. And then you take you apply the CTF and the shifts. Okay, so, I guess took me

Ellen Zhong 52:27

so long to figure out how to parse all those files.

Benjamin Himes 52:31

And then, even once you figure out how to parse them, everyone's conventions are a tiny bit different.

Ellen Zhong 52:36

Oh, my God, holy crap,

Benjamin Himes 52:38

like the bane of my existence. You wouldn't think to be I mean, you know, Euler angles have lots of different ways that we represented, but it's so much more of a headache than, yeah, I feel your pain.

Ellen Zhong 52:49

Oh, man in the beginning.

Benjamin Himes 52:53

For the longest time, I had three different colored pens taped to my desk, well taped together on my desk, to make a little coordinate system that I could rotate around to try and like, visualize what people are doing. It's goofy,

Ellen Zhong 53:04

but my worst nightmare is that the developers a lot of these software packages will just change their representation.

Benjamin Himes 53:13

Right? Well, so you didn't, you didn't catch that rely on switch from pixels to angstrom coordinates at some point, right? I think that was, that might have been a little while ago. But

E Ellen Zhong 53:22

yeah, luckily, I'm so glad and definitely makes sense to do this. But they changed like the heading and the star file. So like, at least you know that it's different.

B Benjamin Himes 53:34

Okay, so. So we flipped back and talked about the encoder a bit, you go through, and then now you get a representation of the conformational heterogeneity. Which is, that's sort of the main output, right?

E Ellen Zhong 53:49

Yeah. So the main output is this like low dimensional representation representation of the particle dataset. And then also like the train girl, the train decoder, which parameterize? Is this, like ensemble of 3d density amounts?

B Benjamin Himes 54:05

So is there anything else you want to cover there? Do you want to maybe switch to talking about some of the results that you've seen on real data?

E Ellen Zhong 54:13

Yeah, I think the only thing that I would want to mention is that for so long, like the goal for me was to be able to model continuous heterogeneity and to model this like distribution structures. But now that we've done this, and we've showed this on a lot of real datasets, now the question is, okay, now, like, what do we do with this ensemble of structures, and it's like a complex nonlinear, maybe like 10 dimensional space, and now there's this like, new data analysis problem of like, okay, how do we even visualize this entire distribution of structures in a way that like, we can understand that like we humans can understand which I think we really like simple one dimensional motion. So how can we summarize things first, effectively?

B Benjamin Himes 55:05

Well, that actually speaks to something that's sort of the core of my structural biology journey. You know, I did a, spent a few years in a crystallography lab as an undergrad. And that's sort of set me on my way. But one of the things that I came out of this experience with was, well, that's great, we've got this high resolution crystal structure. Now what? So I know you can look at it. And there's particularly now with the way MD shaped up, you can do some really cool things. But a lot of it was just, you know, looking at the models and using some sort of chemical intuition to kind of decide, well, maybe this informs its function, or you know, an enzymatic activity or something. So understanding what to do with that structure information. To me, it was sort of the big, what do we do next.

E Ellen Zhong 55:48

And it's still very mysterious, as like a non structural biologist of like, oh, like, you can glean so much insight from this 3d structure. It's very impressive.

B Benjamin Himes 56:01

So you have a problem now, where you got, you could almost think of it like a trajectory, and what to do with that information. So before maybe before you, if you want to talk a little bit about what you think you might do with that. One question I would have for you is, do you after looking at results? Do you think there is specifically this continuous low energy landscape that, you know, basically, all confirmations could be sampled? What do you think it's, you know, a very, you know, low and flat thing where there's a lot of states, but at some point, there's still a discrete number of states that are actually ever observed.

E Ellen Zhong 56:41

Yeah, so I think answering this question empirically, we've seen like the whole gamut of different like behaviors, these different particles, I think it's very dataset dependent. So in the L 17, or the assembling ribosome data set, we call it L17. Because the like L17. Protein has been depleted, to like be able to capture the whole set of intermediates from assembly. But anyway,

B Benjamin Himes 57:09

if you want to talk about a field with a ton of jargon, ribosomes, and then if you flip between bacteria and yeast, it's just

E Ellen Zhong 57:16

completely different. Okay, so the bacterial assembling ribosome data set. There, we saw like this, we saw like a very clustered latent space with different assembly intermediates, for from the different clusters. And so and the different clusters were very distinct. And actually, the late representation was useful for you in discovering new assembly states, because we saw like, oh, there's this like, small cluster that's peripheral to this larger one. And in the 3d classification, it was misclassified into this other state. But if we actually generate the structure using our decoder at this, like, teeny little cluster, we see something that's believable. So like, yes, our prior knowledge has like informed us that this cluster is real. But it lends itself to this, like unsupervised discovery of the states, in other datasets is not as like satisfying, because it's just like a continuous blob, that becomes hard to analyze. And so I was kind of alluding to this before, but if we had ways of better timing, so we have our decoder now, but actually, in the whole, like VA framework, the decoder is not like, it's not tasked with making a human integral representation at all. And so in many cases, especially when the ensemble of confirmations is more flat, like more evenly distributed amongst a lot of different states, then we see like a very flat representation in the latent space. And while we see like a ton of really interesting heterogeneity in the structures that we get out, it becomes very hard to analyze this low dimensional representation. So are there better ways you can do moving forward to like, analyze this distribution of states? We couldn't quite tackle this in this one manuscript, but I think that's like a really interesting line of follow up work.

Benjamin Himes 59:15

Trying to think so. Sorry, I'm kind of drifted off onto a... I thought there. So if we think the best way to assess so I guess you're working with datasets that in the case of the L 17. Is it's arrested in some particular set of confirmations? Or is it just it's a purified ribosome? Is it from a lysate? How did it get into the ice? I guess?

Ellen Zhong 59:42

I see. Um, so that was from a lysate.

Benjamin Himes 59:45

Okay, should be sampling a lot of fun stuff.

Ellen Zhong 59:48

So sampling a lot of confirmations, although I think there is some amount of time dependent so it's definitely not at equilibrium. Like Once the sample is, I guess prepared, then it's frozen at a certain time afterwards so that we we don't like. So the assembly intermediates don't fall apart or something like that. Right? So in that case, the snapshot from cryoEM is not exactly an equilibrium ensemble, it's just like a snapshot of the assembly process. Ideally, that's the goal, at least. In other cases, I guess it's very dataset dependent, or just sample preparation dependent of like, what did you do to get your aquarium sample? Is this the actual, quote, equilibrium? Ensemble? Or are you doing some kind of like time resolved or fast kinetic or freezing it in some state? And so I think the nice thing about cryoDrgn is that we can analyze the data from all these different regimes. We're not like imposing the presence of discrete states, like in greedy classification.

Benjamin Himes 1:01:03

So just trying to think then, if you say, I mean, that's a pretty complex mixture in itself, if it's purified from the lysate. In that, when you look at that data, are there other particles very close by or even overlapping? Or is it still fairly dilute? From lysate?

Ellen Zhong 1:01:24

Ellen Zhong 1:01:24

Oh, yeah, this is definitely something that could be an issue, but it's fairly concentrated. But I think it's Yeah, so it's fairly concentrated, but we didn't actually see an issue and the results from cryoDrgn . In other cases, if the particles are too concentrated, then the encoder gets confused, because there are some images where there's one particle and there are some images where there's one particle plus a few more in the corners. And so then it's like, what is this heterogeneity, let's put the ones with like, four particles, and over here, and let's put the one with like one particle and over here. And so you actually see this, or you see a very well, and latent space, we also see this in the generated structures, where like, some particles will just have these like blobs on the side of them. So I think that, right, it might not be as big of an issue and like the voxel based approaches, because you just like average this out, or you just like mass goes out, so you don't even consider it. But in this case, like the encoders, almost too powerful, like sees all sources of heterogeneity. And so it could be a good thing, because now you can like filter your data set based on that, but it also like interferes with what you really want to learn, which is the structural heterogeneity of the particular complex.

Benjamin Himes 1:02:45

So I mean, even though the Oracle sees all I mean, if you pre processed it, like if you mask it in real space before transforming, do you think that? I mean, maybe that would help a little bit, but I guess not with particles on the side anyway?

Ellen Zhong 1:03:01

Yeah, that's a great idea. And I think it might even be really impactful for removing like the micelle in membrane protein complexes, for example. Or just like even doing the real space masking around the corners of images. Could that help there? Probably, I haven't really, like there's a lot of empirical questions that you could ask. And maybe an interesting follow up would be to even look at more real datasets and ask like, all sorts of these kinds of questions like, do we see issues because of this or not? But the real space signal subtraction, just signal subtracted images. I thought that that would introduce a lot of artifacts. And I actually haven't tried it myself. But there are some users that have told me that it like works really well. So I'm like, Oh, great.

Benjamin Himes 1:03:56

Maybe worth giving a shot. So another technical detail I was wondering about So you mentioned that instead of using the full complex representation of the Fourier transform with the object using Hartley transform, is that right?

Ellen Zhong 1:04:13

Oh, yes. Um, yeah, that's, that's correct. And it's a little bit more just for convenience sake. The Hartley transform is very, very similar to the Fourier transform, instead of like cosine plus i*sine it's cosine plus sine as the kernel when you're doing the Fourier transform. And I guess like what it ends up the right the like characteristic in the Hartley transform that makes it really nice for these neural network training is that instead of having to keep track of the real and imaginary components, we just need to keep track of like one value. So instead of like a complex number, we have a real number as the Hartley transform of have an image. And so it's mostly just for convenience sake. So now like instead of having to keep track of like an NxNx2 by two array of values for a given image, now we just have to keep track of it an NxN.

B

Benjamin Himes 1:05:16

So is that are you losing information, though? I mean, if you have to keep track of half as many values, they have to go somewhere?

E

Ellen Zhong 1:05:24

Ah, yeah, that's a great. Great observation. So you don't have you don't actually lose information because the Fourier transform has doubled information in some ways for because we're like a real valid image, the, what's the word here, the half of the image is a complex conjugate of the other half. And so for a image, you really only need to keep track of like half of the images, complex numbers. And so the amount of information is actually the same. In the real representation, or the free representation, hardly representation, just from like a programming or implementation standpoint, it's easier to learn the, like, an n by n array instead of a n divided by two by and buy to array.

B

Benjamin Himes 1:06:16

Okay. So keeping a two dimensional instead of what's effectively three dimensional, makes it a little easier. Yep. Interesting. I will have to, you know, it's just something that piqued my interest. And I have to go read about Hartley transforms a little bit.

E

Ellen Zhong 1:06:30

Yeah, if you are interested, there is like a particular there is like a quirk that I spent, like some amount of time looking into for cryoDrgn . So there's actually a flag in the software of changing the domain, like, do we want the heartleaf space representation or the phrase base representation. And I've actually switched back and forth couple of times in the code base. And this is before releasing it. So it's been like very stable, since releasing, but what happens in the Fourier representation is actually slower to train. So in the end, this is more because of crowd dragons architecture. So in the Fourier, sorry, in the Hartley representation, we have to reconstruct all n squared images. Whereas in the sorry, did I say that right? In the Hartley representation, we have to reconstruct all n squared pixels. But in the Fourier space representation, we have to reconstruct half as many pixels, we just have to predict two values per pixel, right. And because of the architecture of the neural network, it's actually twice as fast to do it in the larger format, instead of, the former. And so it gives you a major speed up in some ways, like instead of something taking six hours, it takes 12 hours, or something to be pulled out hours, take six hours, there is like a particular, like, quirk, which is if you which is that the neural network never learns like the boundary between the two halves. And that can lead to some weird artifacts. But more so if you're doing pose estimation. Anyway, I leave that as a flag. And I haven't really talked, I have, I haven't had this is like the first time I've been able to tell anybody about this.

B Benjamin Himes 1:08:33

Heard first here on cryo2go,

E Ellen Zhong 1:08:37

hah, Oh, yeah. So yeah, that was kind of an interesting, just like finding when I was developing the architecture. Very cool.

B Benjamin Himes 1:08:52

Alright, so we talked a little bit then about, you know, what you see coming down the pipeline a little bit in terms of where something like crowd dragon or crowd tracking specifically can go. So how, you know, this is a pretty big result. So I assume, are you kind of toward the end of your PhD? Or where are you at in that whole journey?

E Ellen Zhong 1:09:11

Yeah, I think I am. I'm starting to think about what I want to do next. And I still have to do more like existential soul searching, and deciding, like, as we talked about, in the beginning, like I've always felt very in the middle between, like machine learning and cryoEM, or more generally, like biology. And so I think I have to decide, maybe I don't have to decide but I have to, like do a little bit of soul searching in like, which side of things do I like better or to look into? And I think I just have to look into like, what are opportunities like what are the next steps? That's not something I'm not super familiar with. Actually, at this point. I've been too focused on like, working on the software and like working on the papers,

B Benjamin Himes 1:10:02

it's a soul searching. But still in terms of sticking with science, and academia.

E Ellen Zhong 1:10:09

I would love, I would definitely love to stick with science. I'm not sure about academia, that is one of the like, classical questions that I think is on the mind of a grad student is like, oh, do I want to see an academia do a postdoc? Or do I want to go into industry?

B Benjamin Himes 1:10:22

Right, Particularly someone with a background? Do you have a little bit of marketability there?

E

Ellen Zhong 1:10:28

Right? Yeah, industry is very cushy, I imagine if you know how to program and stuff like that. But there's just so many trade offs, it's like very hard to decide anything. I think I have to, like knowing myself, I would probably just, like, just see what's out there, and then applied everything. And then based on the opportunities, then like, filtered down into, like, once I have like concrete opportunities that I like, have actually done research into and like, know more about than decide. But at this point, I have no idea.

B

Benjamin Himes 1:11:06

So the one thing that I can offer, from my own experience, there is apply to things, that's great. But just email people that you, you know, obviously takes finding someone that you would be interested in working with. But you know, people like an email, especially if you can make some, you know, simple case for you know, here's who I am and why I care. And I guess maybe, maybe it doesn't work right now. Because flying out and meeting them is trickier. But, you know, just getting to sit down and talk with people really can. I know it influence my decisions a lot. Obviously, your resources are important, but you know, making sure you're in a people environment. That's good is huge.

E

Ellen Zhong 1:11:47

Yeah. How do you feel now leaving Janelia like I was at Janelia, for this women in computational biology conference last fall, or over a year ago, for less than nine months ago. And I was totally blown away by like, how like, crazy that place is. beautiful it is and everything.

B

Benjamin Himes 1:12:07

So I'm still coming down a little. I knew it, I knew it would be a little bit of a challenge to move. You know, I won't say too many things. Because UMass is also you know, it's a very nice campus, and the people are great. But, you know, the building the landscape and the environment, you know, just like walking into work every day, you know, from well I mean, first off, my parking space was, you know, 30 feet from my desk in an underground warm garage, which was awesome. I'm a little terrified of the New England winter and having to walk 10 minutes, which still isn't so bad. But I think the thing that I'm probably missed the most, and again, nothing against UMass, but you know, a big university, there's more bureaucracy, whereas at Janelia it was kind of run like a hybrid between an academic institution and more of a business. So everyone's role was very clearly defined in terms of all the non scientists, I don't want to say support staff because they had their own stuff to do. But if you needed something to be fixed, you only had to email one person. And if they weren't the person do, they would find someone else to take care of it for you. So I got very spoiled. So you know, aside from the aesthetics, yeah, just the functionality. In there was the goal of the place to remove that burden from people doing science. So they could just do science, some still, I guess, learning how to do all the other stuff again.

Ellen Zhong 1:13:30

Yeah, I remember that very distinctly. When I was working at like a tech company. It was just, like, especially so I did research and undergrad in like, related field. This is right. So

Benjamin Himes 1:13:43

Oh this is right, you were at DE shaw?

Ellen Zhong 1:13:46

Yep. Yeah. So I was working at like tech company, aka, like, private research group for molecular dynamics, supercomputers from molecular dynamic simulations of protein structure. That often gets confused with like a hedge fund research, like division. But yeah, so I was like, do your research. And there it was, as you described, like teams for the different projects. And there's like a whole software team, there was a whole infrastructure team, the resources there were phenomenal. And like the people who were doing kind of like the infrastructure, human development were like, like, way ahead of their time in terms of setting everything up. So it was very seamless. And um, yeah, I missed that.

Benjamin Himes 1:14:35

Yeah, but I mean, I guess at the very least, you know, it exists and that kind of can influence your decision a little bit, too. Yeah. Well, there's no easy decisions, that's for sure. But it sounds like you're coming from a strong enough foundation that I think you'll you'll end up just fine.

Ellen Zhong 1:14:52

The Coronavirus pandemic has also, like thrown a lot of uncertainty into the whole like hearing stuff and like, what? What's out there? And so I'm trying to just like, take it kind of, or like not try to, like plan out too far in advance.

Benjamin Himes 1:15:10

Right? Well, that's so that's something I was, you know, before it had I was, you know, trying to decide I had an opportunity to start my own group at a NIST, which is, you know, a government agency, which certainly had some nice perks, it would be corps funded and be my own small group. But there was also some drawbacks, you know, it's like in this windowless hole in the ground, which speaking of going from Janelia, to somewhere different. And I, you know, for other reasons, I'm still very happy that I went throughout I did, but I can't imagine trying to hire new people right now. Which is super critical, you know, in your first few years as an assistant professor. Just be super tough. Yeah, it's a good time to sit back and have those existential thoughts in a cup of tea and winter, I guess.

Ellen Zhong 1:15:59

Yep. Agreed.

Benjamin Himes 1:16:03

I like it. I like it. I'm gonna try and make that happen. Cool.

Ellen Zhong 1:16:07

Cool. All right. We have to pop off to this other zoom call. Oh, yeah. Wow. Yeah, that time went by really quickly.

Benjamin Himes 1:16:17

Well, it's great talking to you. I'm glad I finally tracked you down. I appreciate you taking the time.

Ellen Zhong 1:16:21

Yeah, thank you so much. And it's probably better to talk now versus like, last, like a few months ago, because I think now I have a much better understanding of like, how cryoDrgn is performs and things like that.

Benjamin Himes 1:16:35

Yeah. All right. Well, I will get back to you with both those follow up questions and whether or not I run into trouble trying to use cryoDrgn myself. Cool. Good. All right. Talk to you. Thank you. All right. That was episode seven of cryo2go. If you made it this far. Appreciate you. Hope you enjoy the conversation that Ellen and I had. If you could take a minute to like one or subscribe to the show wherever you're getting your podcast that definitely helps other people. Have it show up in their feed and maybe appreciate and enjoy something new that they weren't expecting. So until next time, be easy and be good to each other.