

Nonlinear PCA for photonic lantern wavefront sensing

Aditya Sengupta

The purpose of doing linear PCA is to find groups of lantern ports that tend to move together, as that is very likely the bottleneck in wavefront reconstruction. There are numerous approaches for getting reconstruction arbitrarily good given a faithful representation of the output, and I believe that the limiting factor on lantern wavefront reconstruction performance is what modes tend to be mapped into what other modes or tend to be read into other modes as a function of the forward process. In the presence of shot noise, it is possible that one phase screen gets mapped into another one, potentially causing a sign inversion. I'd like to be able to map out the cases where that is likely to happen for any given manner of design, so that we can move towards optimality using heuristics without having to make a tractable model. Optimization over the whole design space.

The NLPCA approach I've been taking involves using the reconstruction neural network and shrinking the size of the limiting layer, repeating this at a variety of F numbers. At some point, the limiting layer size tells us something about the dimensionality of the space that we're living in. So far, with the design I've been using throughout this, I have found that the limiting layer size of the optimal F number of 4 or 5 is about 15 for a 19-port lantern. Barnaby suggested I increase the size of the whole network so that I don't have any accidental underfitting error convolved with the dimensional effect I am trying to find. I don't think this ended up affecting the discovered dimensionality much, but it did greatly affect the correlations I've been able to spot.

Even so, I'm not sure this is exactly the right approach because it seems like NLPCA is usually done on the identity matrix just with output data. I think what I've done still answers a question that is relevant, but it's maybe the second thing we should have asked. The first thing is whether these correlations exist in the data, and the second thing is whether that affects overall reconstruction performance in the way that I've been measuring it here.

You can begin to assess the first one just by doing regular PCA on a data matrix of lantern outputs, say 19 by several thousand, and trying to find groups that always move together. These are the features that an interaction matrix approach needs to pick up on and that it's necessary to have inherent to the lantern so that there are linear features to use for reconstruction. But we already know in reality a linear approach is insufficient, plus one of Barnaby's students has results saying a lantern may be very good much further out than we think. and may be useful for things like petaling and low-wind effect. And so that means we are going to need to care about correlations beyond the linear because correlations beyond the linear are what reconstruction approaches like neural networks, Gerchberg-Saxton, and gradient descent are going to care about.

That means we're going to need to find degeneracies in lantern responses that take into account non-linear correlations. This is exactly the role of NLPCA. We need to look at the maps that you see from lantern outputs to what goes into the bottleneck layer and that needs to be attempting to make an identity operation.

From there, we should be concerned about how this bottleneck impacts reconstruction quality and what correlations we can end up seeing. That is the step that I've already done.

So really the bit that is left over is the relatively easy bit. I need to try running regular PCA on a set of lantern outputs that I've generated for the sake of doing this neural network training.

Then I need to repeat this with NLPCA and figure out how to extract not just the dimensionality of the space, but also the shape of these nonlinear correlations that the network needs to pick up on to do good reconstruction. And that is going to allow me to make some statement about eventual limiting reconstruction performance.

(An open question I have about all this is: What is a good way of drawing a set of test phase screens that is going to be representative of what we're trying to actually do?

In a way, I'm thinking of Jon's work that he showed me recently here. If we expect that NCPAs are going to be relatively small, it doesn't really matter if we have multiple possible phase screens that map to about the same lantern output because we know we are going to stay relatively local to where we were before.

So I would like some clever way of narrowing down the question, I guess.)

Here's the 16 principal components from the training set:

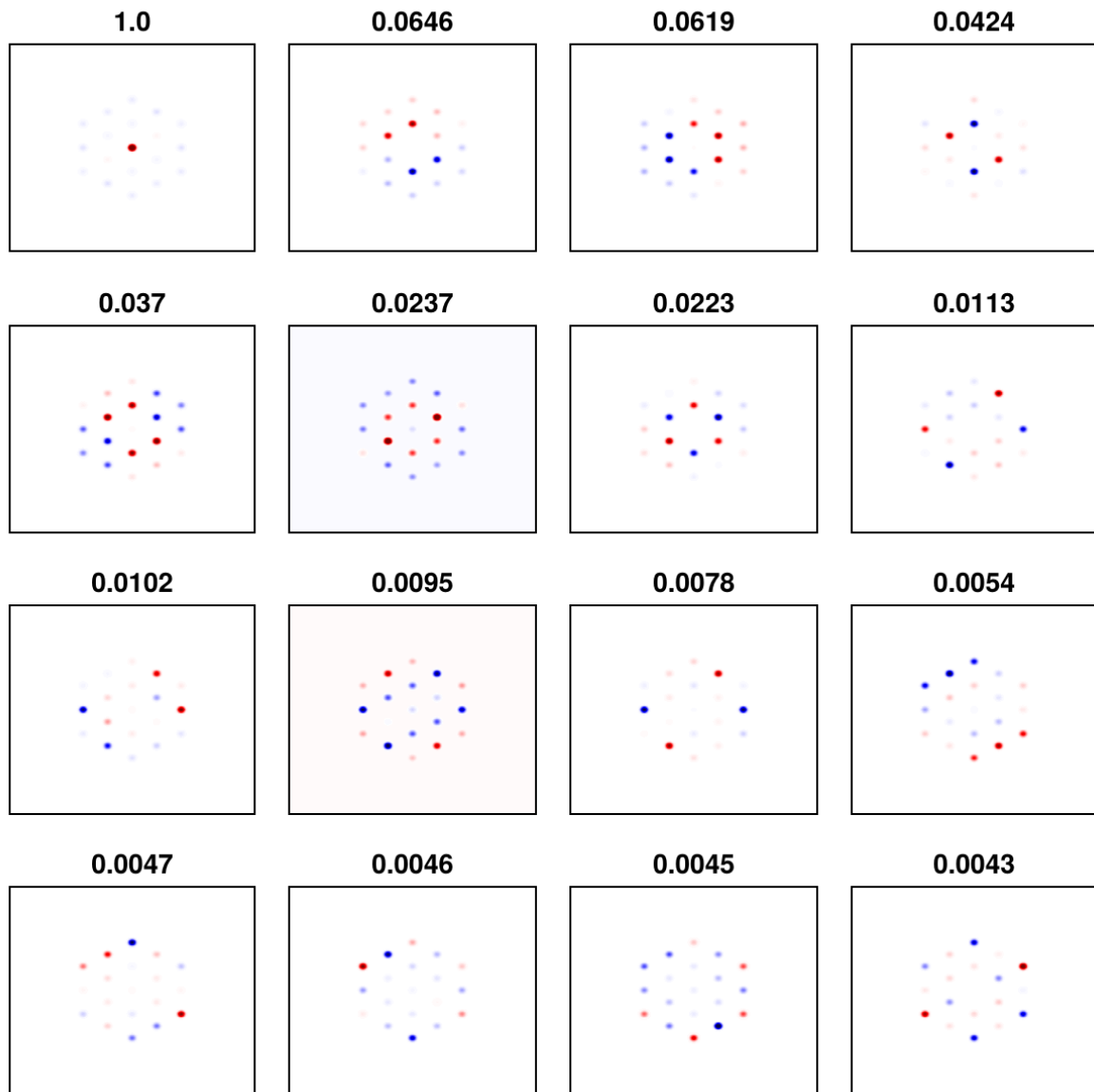


Figure 1: The principal components of the simulated PL response.

And here's the results of my new NLPCA run with the larger network.

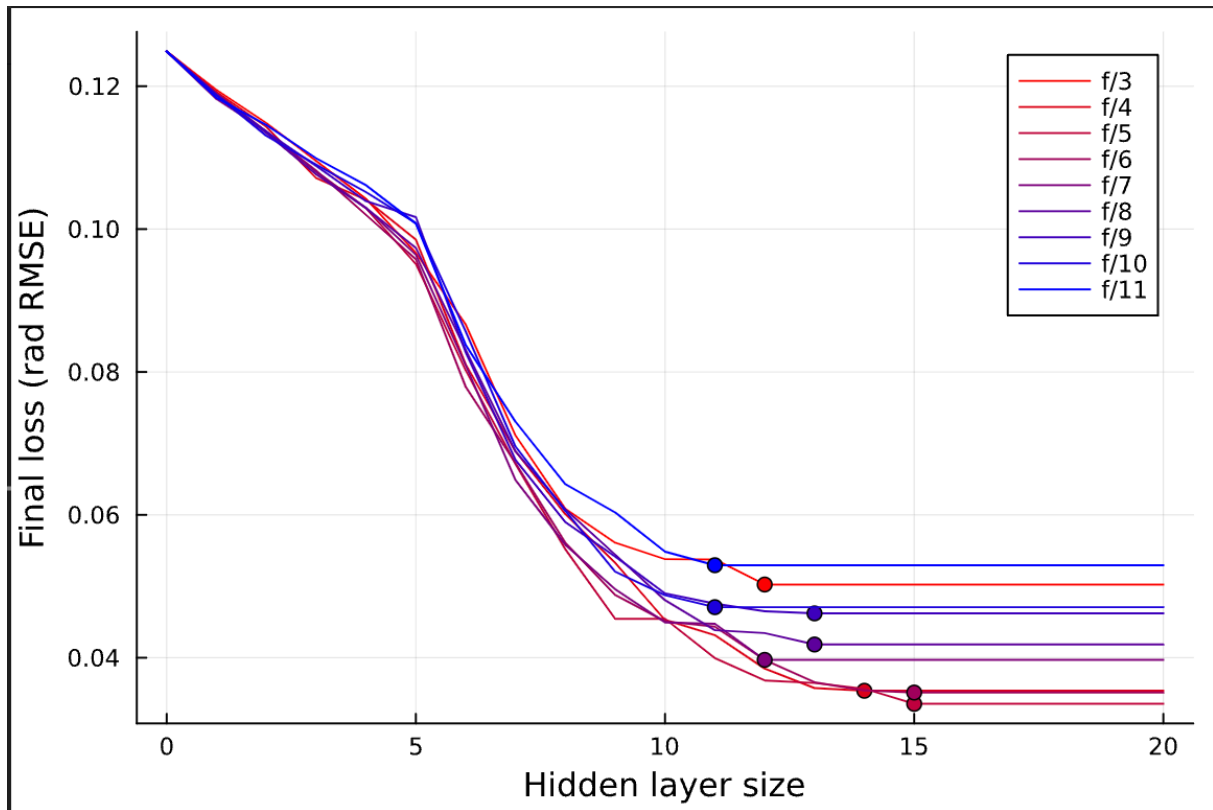


Figure 2: The losses achieved at each f-number and hidden layer size.

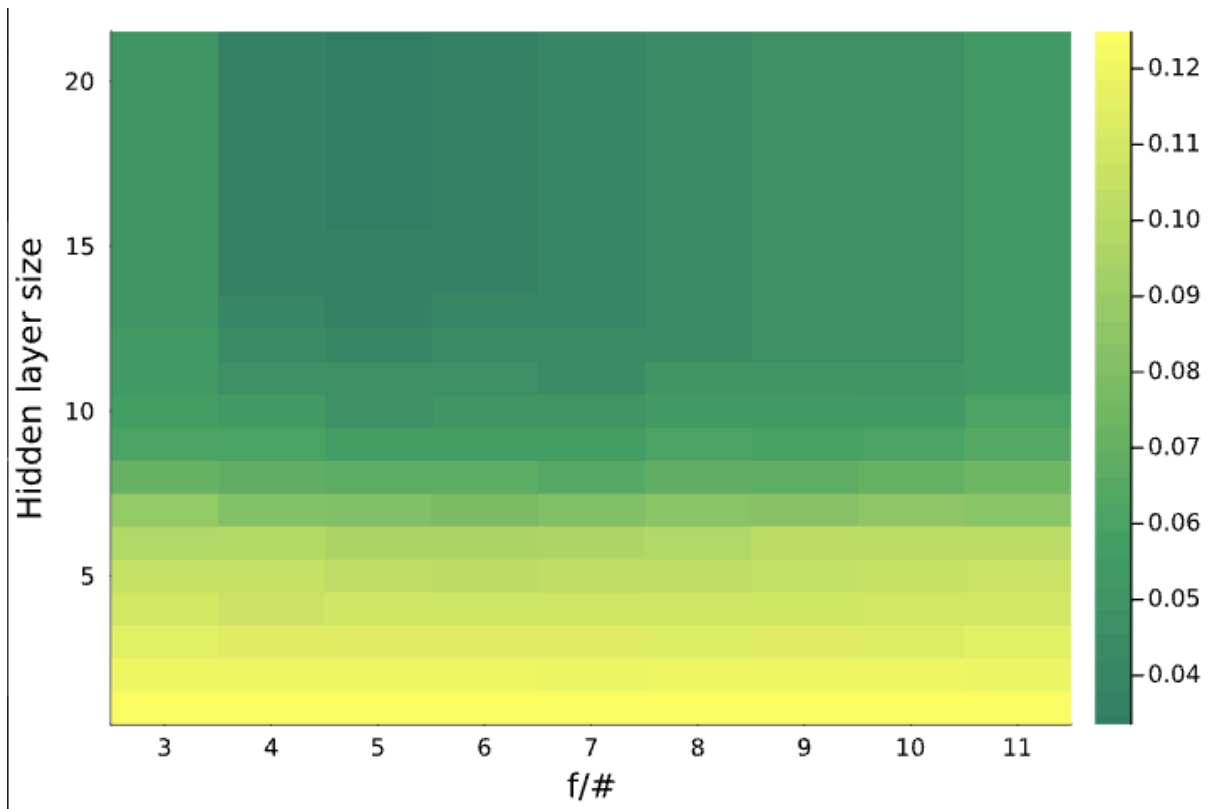


Figure 3: The losses achieved at each f-number and hidden layer size.

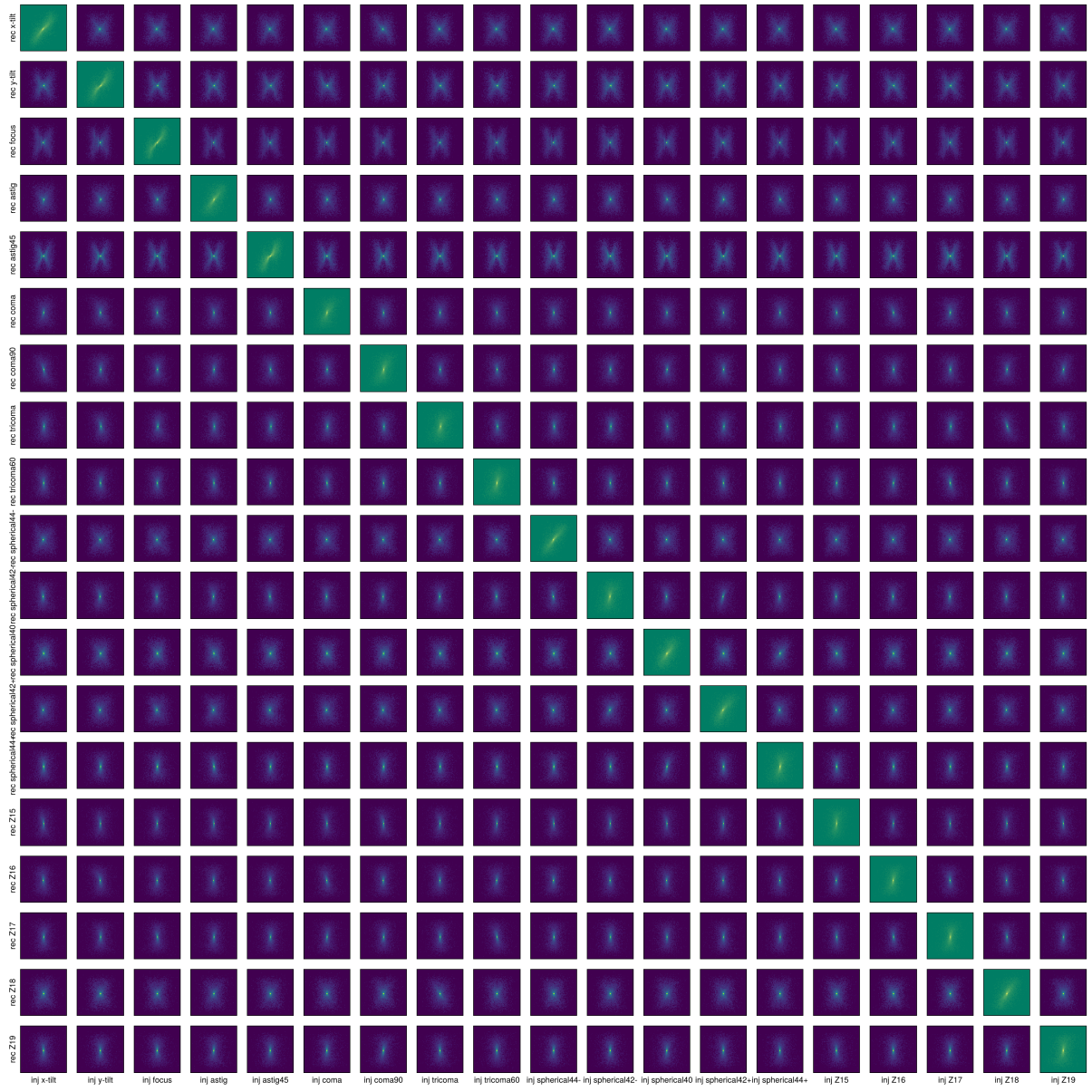


Figure 4: Nonlinear injection/recovery between each pair of Zernikes.

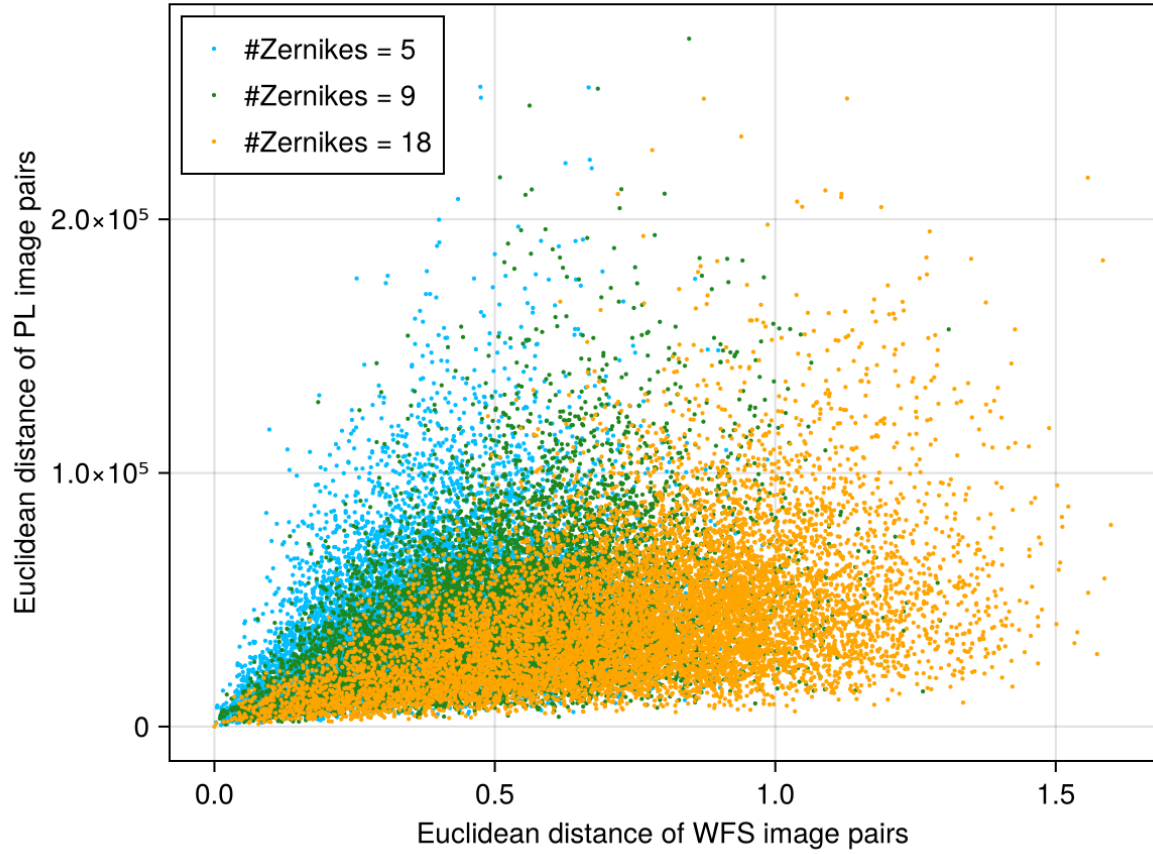


Figure 5: Euclidean distance “plume” plot showing correlation between distance in phase vs. PL output spaces.

I’m plotting this and the next ones with 5, 9, and 18. 5 for (tilt, focus, astig); 9 to match Norris+20 and my SPIE paper; 18 for the limit.

This one’s interesting! It shows that you can have far apart WFS responses with similar PL outputs, but you can’t have far apart PL outputs resulting from similar WFS responses. This is to be expected: you have some loss of information going into the PL, so this is what a projection to a smaller space should look like. The question I’m interested in from here is, does this all show up when I plot Euclidean distance in the NLPCA embedding/mapping space?

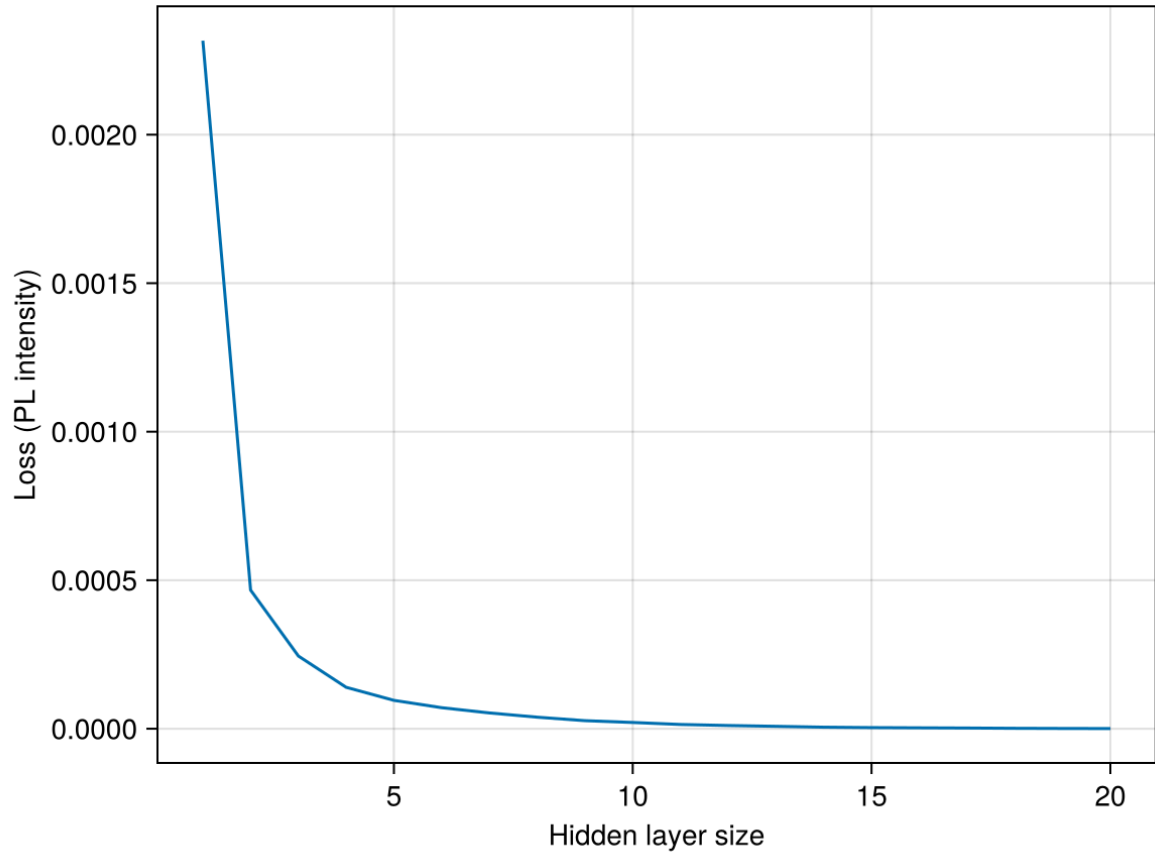


Figure 6: Losses from running NLPCA on the identity operation on PL outputs.

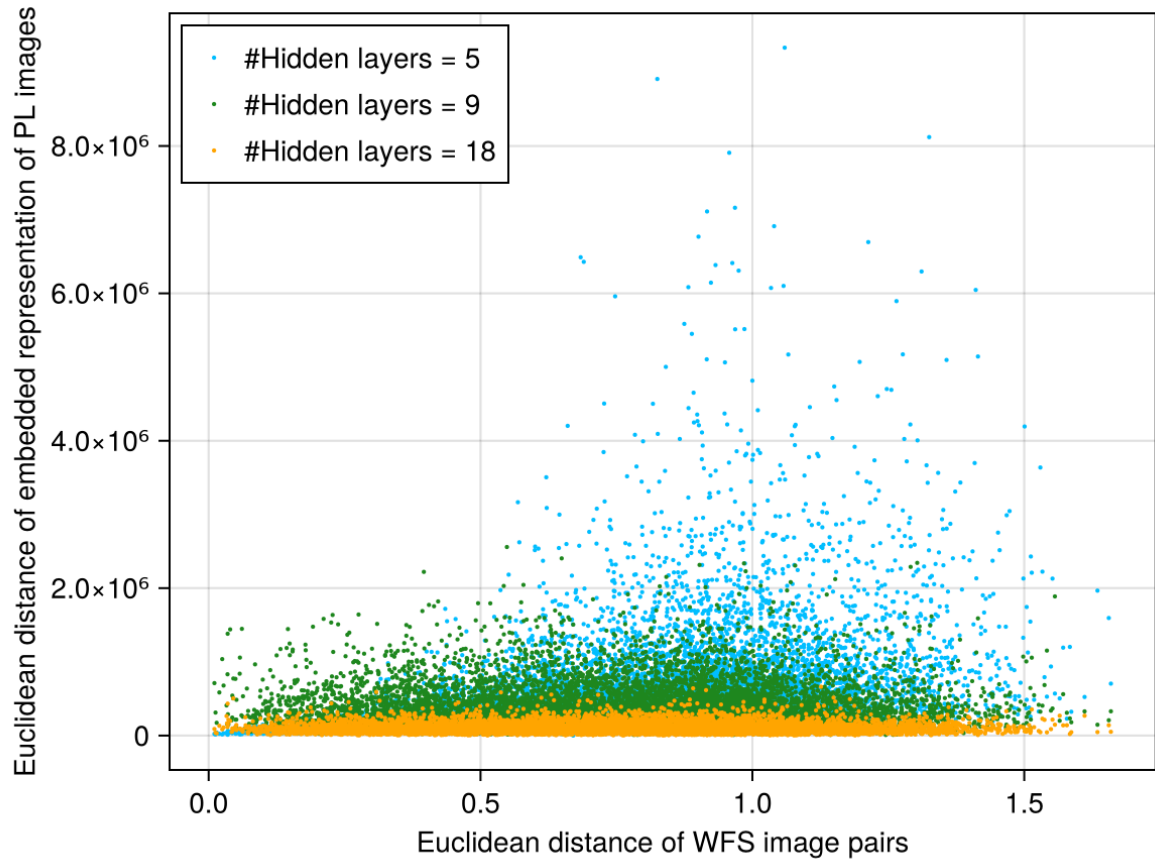


Figure 7: Euclidean distance “plume” plot showing correlation between distance in phase vs. distance in PL embedding.