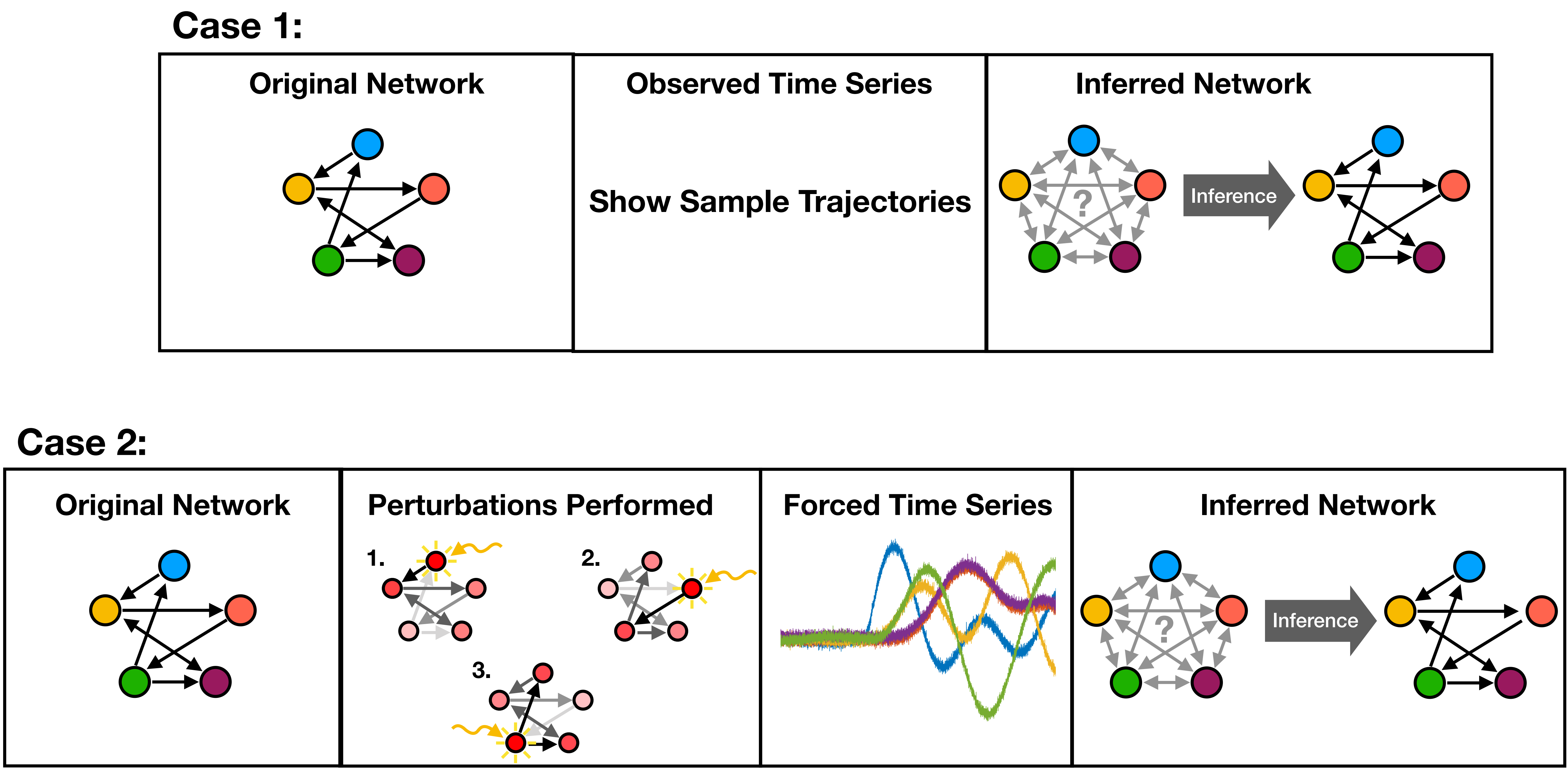
Figure 1

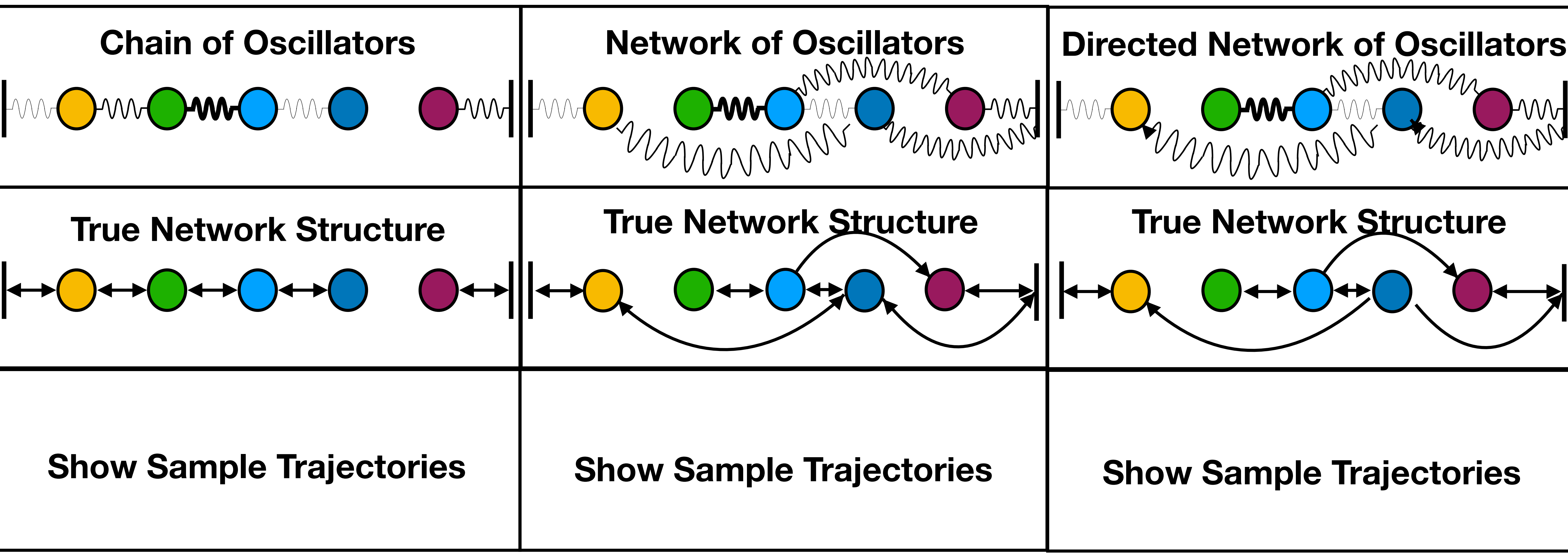


This figure will show the reader the storyline of our paper (very similar to Bethany's Figure 1 in her GC paper). We first start with a network of nodes that produces time series data. We get noisy time series observations. In the first case, these time series observations are given to us. In the second case, we are allowed to intervene and force the network. In both these scenarios, we would like to reconstruct the network that produces this data. The reader will become familiar with the concept of a network, time series data, perturbation interventions, and network inference from this image.

Figure 2

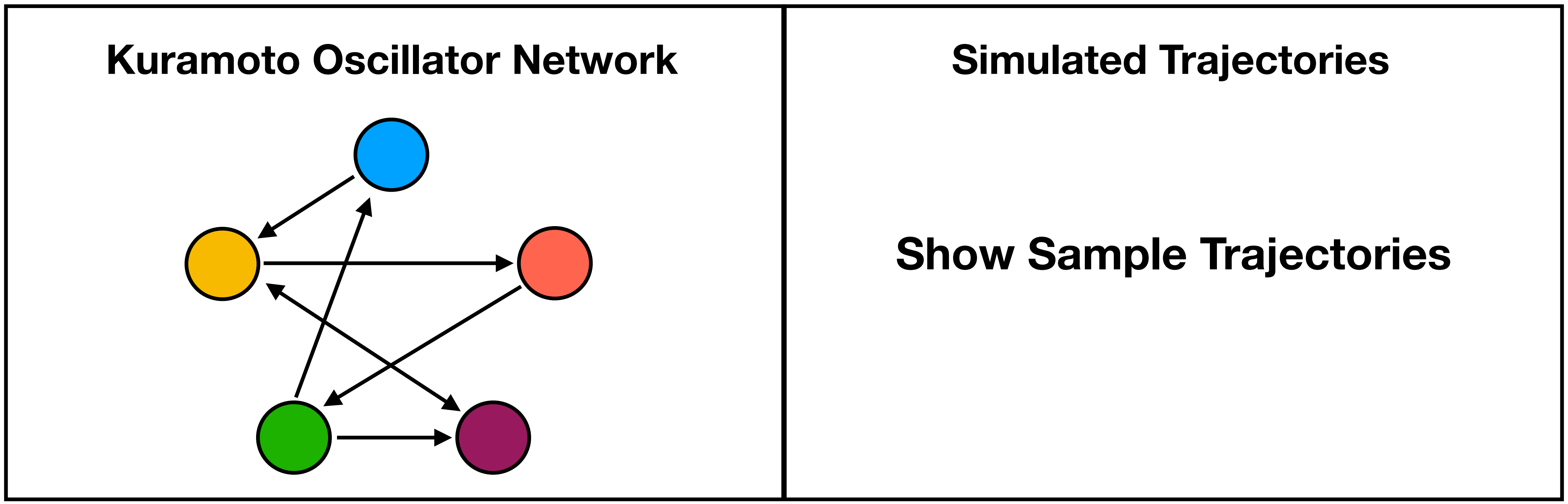
This figure is very cluttered. Will need to think how to simplify/ remove some components.

Need to figure out how to draw directed network with springs, this is a nonphysical system



This figure will show the reader the three type of harmonic oscillator systems in order of increasing complexity: chain of oscillators, network of oscillators with more than nearest neighbor interactions, and a directed network of oscillators. Each column shows a physical picture with springs, the true underlying network structure and a simulation. Helps the reader visualize the data generation process and familiarize them with harmonic oscillator networks.

Figure 3



This figure will show the reader a sample Kuramoto oscillator network and simulated trajectories. Helps the reader visualize the data generation process. Also important to stress here that Kuramoto oscillator networks naturally have directed interactions.

Figure 4

Harmonic — GC

True \ Pred	● ●	●←●	●→●	●↔●
● ●	10	1	2	87
●←●	0	0	1	99
●→●	0	1	0	99
●↔●	1	0	0	99

Kuramoto — GC

True \ Pred	● ●	●←●	●→●	●↔●
● ●	71	10	12	7
●←●	2	3	0	95
●→●	1	0	2	97
●↔●	1	2	0	97

Harmonic — CCM

True \ Pred	● ●	●←●	●→●	●↔●
● ●	1	3	2	94
●←●	99	1	0	0
●→●	93	0	6	1
●↔●	46	1	1	52

Kuramoto — CCM

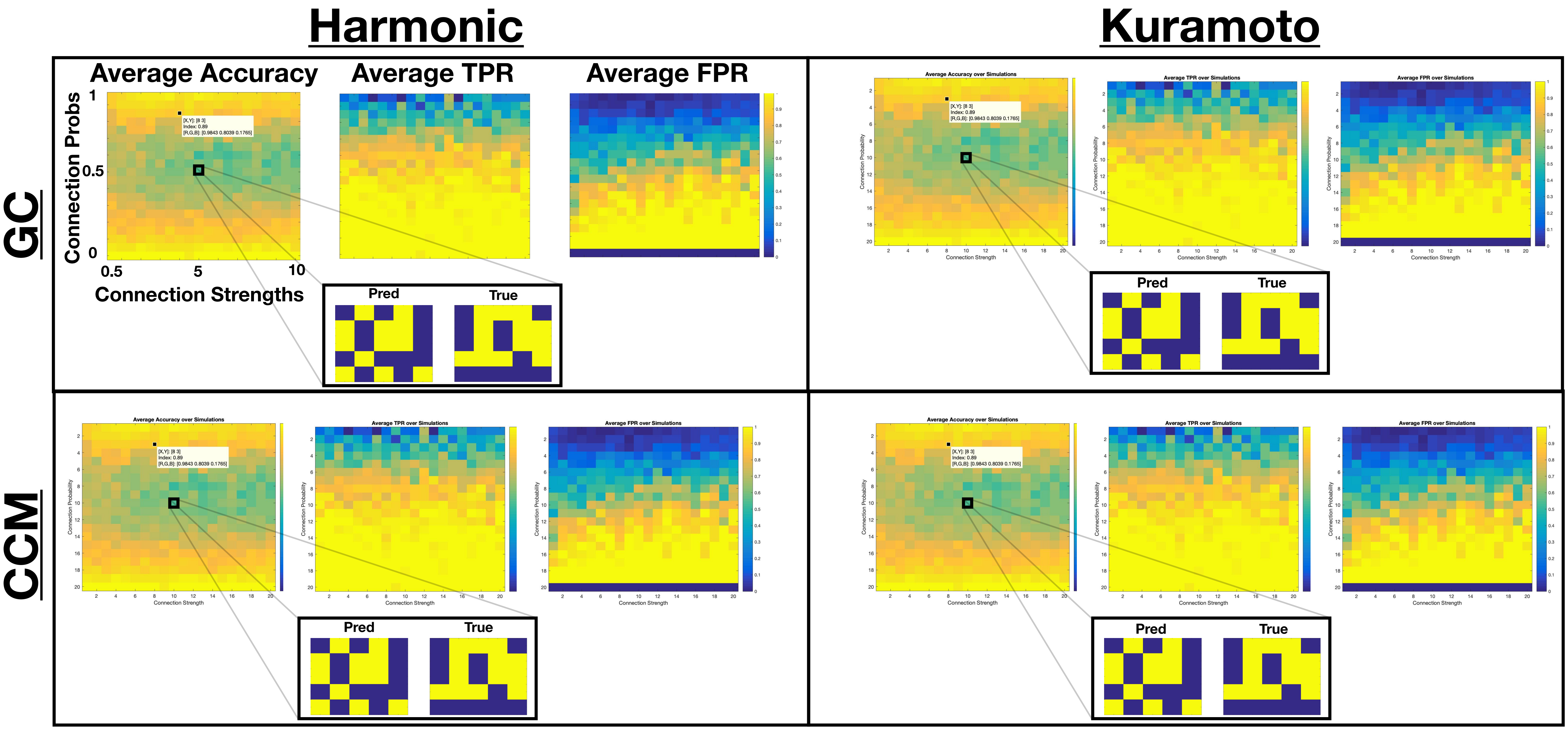
True \ Pred	● ●	●←●	●→●	●↔●
● ●	90	2	8	0
●←●	7	3	0	90
●→●	3	2	2	93
●↔●	1	1	2	96

This figure shows how poorly GC and CCM do on the smallest possible network (two nodes). For every possible connectivity, GC and CCM predict overly sparse or dense matrices. After this, reader might expect the methods to do just as poorly on larger network sizes.

Figure 5

I'll run these same experiments for networks of 3, 4, 5, 10, and maybe 20 nodes. Will probably only show images for the 5 node case because it is a small enough size that you'd expect GC to perform well on (while it actually doesn't).

GC Performance on 5 Node Harmonic and Kuramoto Oscillator Networks with Varying Connection Probabilities and Spring Constants

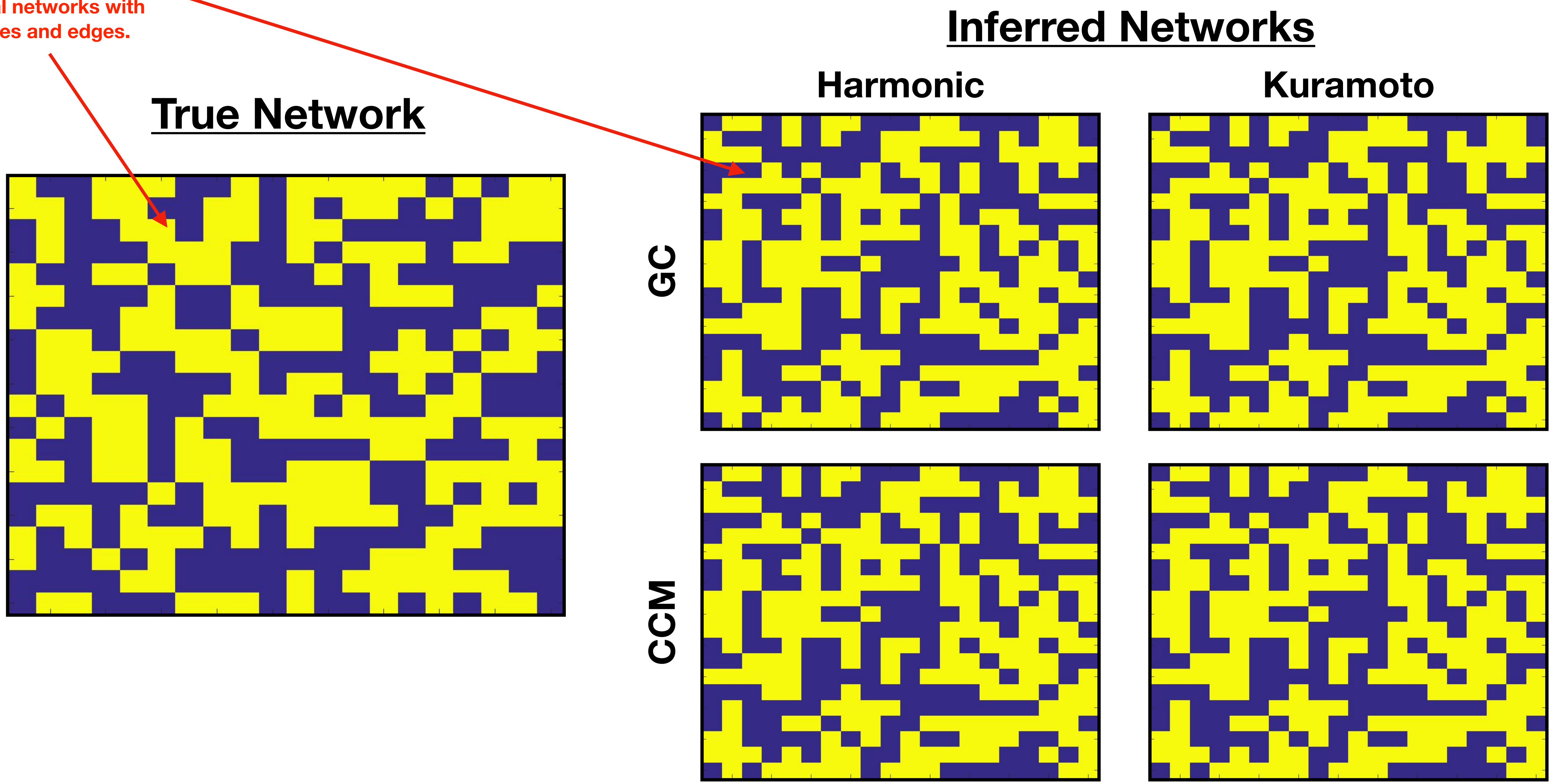


This figure shows GC and CCM performance on harmonic and Kuramoto oscillator networks of 5 nodes. It shows the accuracy, true positive rate, and false positive rate of GC on every 5 node network while varying connection probabilities and spring constant strengths. The reader should see that GC and CCM performance is only good for fully connected or very sparse networks (it's accuracy is about 0.6 in the middle of the accuracy table on image 1). The TPR and FPR plots also support this.

Figure 6

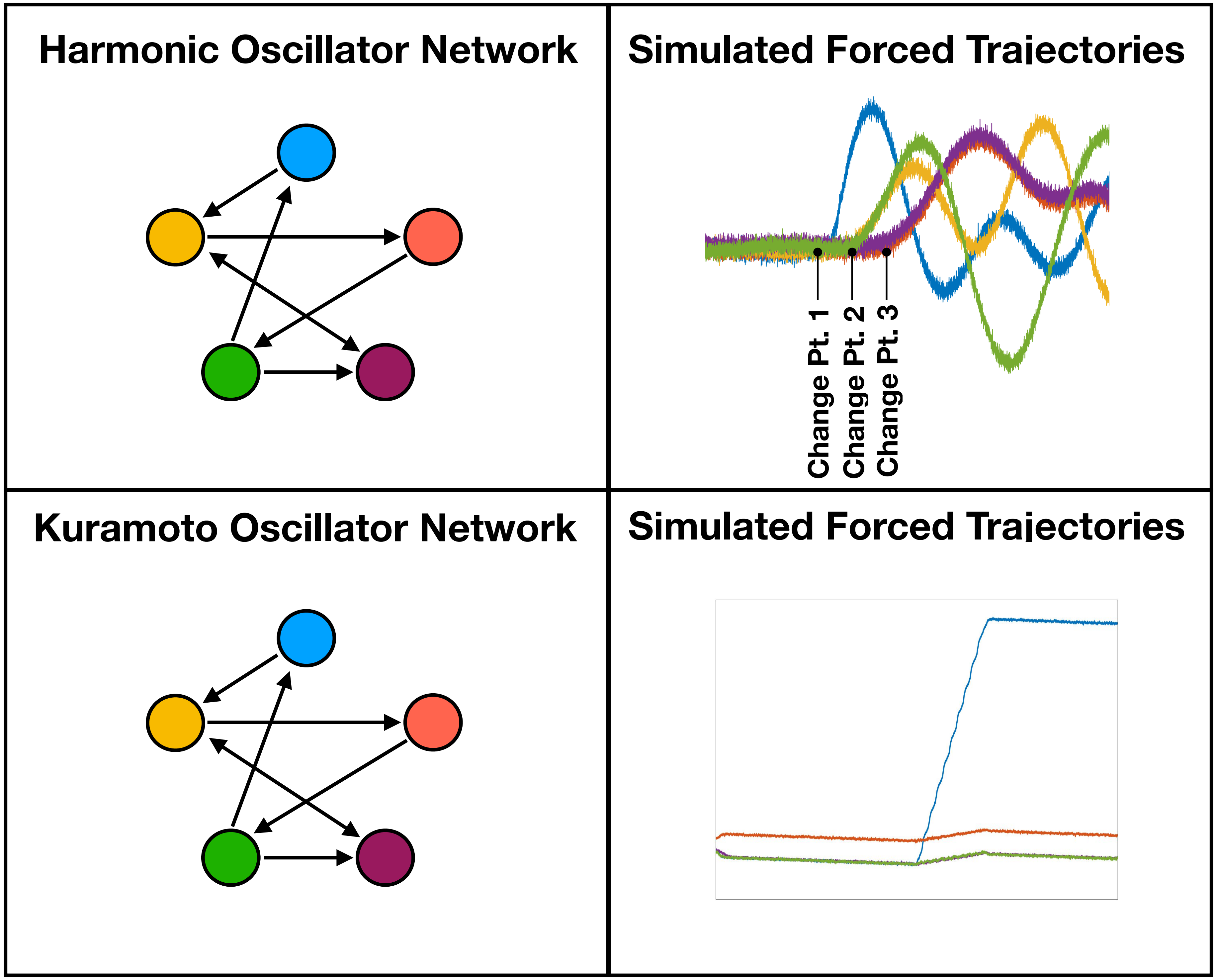
GC and CCM Performance on 20 Node Harmonic and Kuramoto Oscillator Network

Maybe instead of drawing these as adjacency matrices, we draw them as actual networks with nodes and edges.



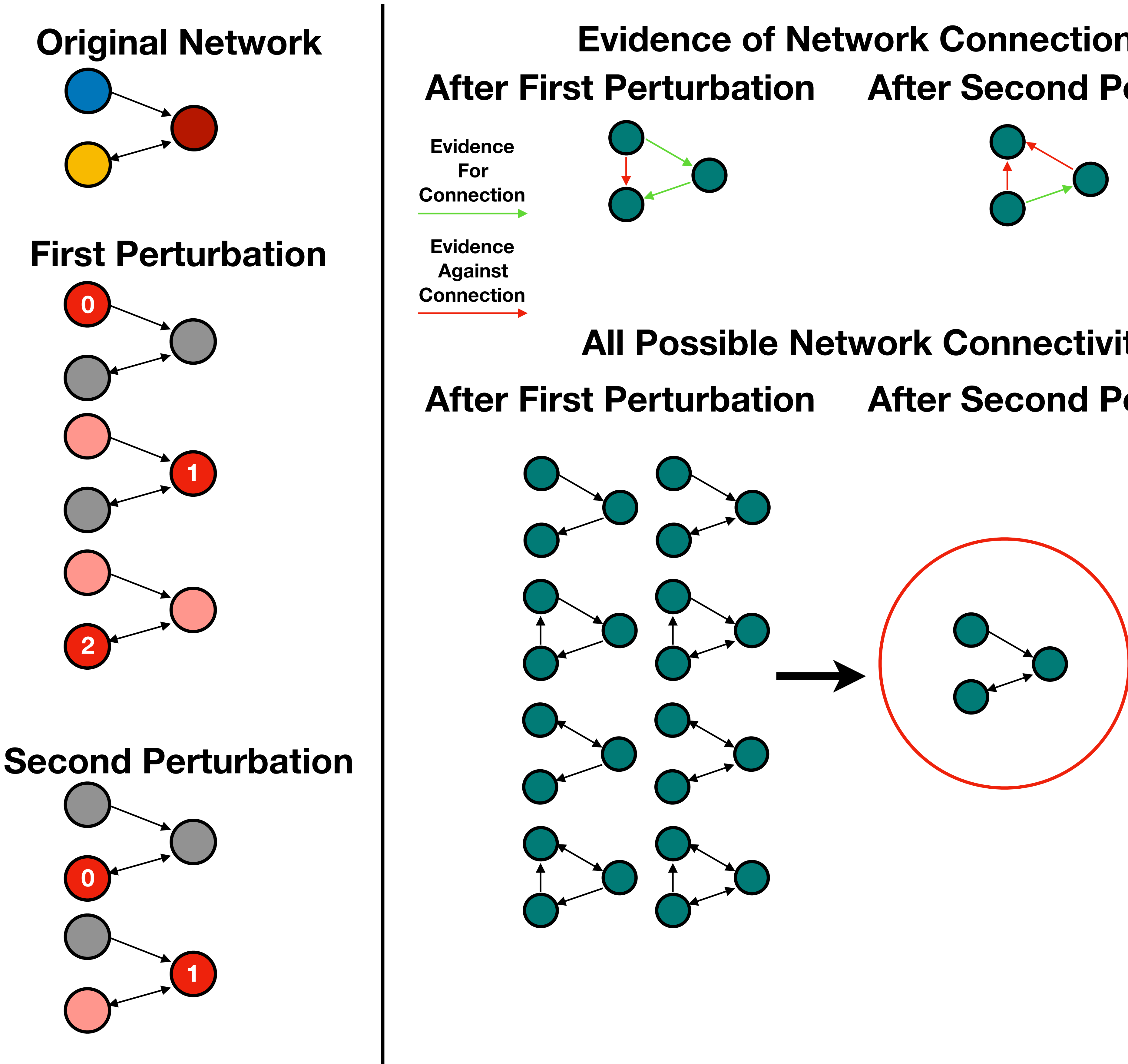
This figure is an example of how GC and CCM do not scale to larger network sizes. They completely fail on a network of size 20 with 0.5 connection probability and connection strengths equal to 1.

Figure 7



This figure is meant to show the reader what perturbed (forced) harmonic and Kuramoto oscillator data looks like. It also acts as a reference to look back to when we discuss the data analysis (correlation and variance statistics) used in the perturbation inference method.

Figure 8



This figure explains how our Perturbation Causal Inference (PCI) algorithm works. With every perturbation, we discard the networks that don't satisfy our perturbation cascade evidence. This picture should show the reader how we can reconstruct a 3 node network with just 2 perturbations. So there is hope that in large networks, a few correctly placed perturbations can give us good inference. Of course, the underlying inference framework is probabilistic (every edge has a probability of being kept), but this simplification is visually more appealing.

Figure 9

Harmonic | Kuramoto

GC

Harmonic

True \ Pred	○ ○	○←○	○→○	○↔○
○ ○	1 / 1	1	2	87
○←○	0	0	1	99
○→○	0	1	0	99
○↔○	1	0	0	99

Kuramoto

True \ Pred	○ ○	○←○	○→○	○↔○
○ ○	71	10	12	7
○←○	2	3	0	95
○→○	1	0	2	97
○↔○	1	2	0	97

CCM

Harmonic

True \ Pred	○ ○	○←○	○→○	○↔○
○ ○	1	3	2	94
○←○	99	1	0	0
○→○	93	0	6	1
○↔○	46	1	1	52

Kuramoto

True \ Pred	○ ○	○←○	○→○	○↔○
○ ○	90	2	8	0
○←○	7	3	0	90
○→○	3	2	2	93
○↔○	1	1	2	96

PCI

Acronym for Perturbation Causal Inference

Harmonic

True \ Pred	○ ○	○←○	○→○	○↔○
○ ○	100	0	0	0
○←○	0	100	0	0
○→○	0	0	100	0
○↔○	0	0	0	100

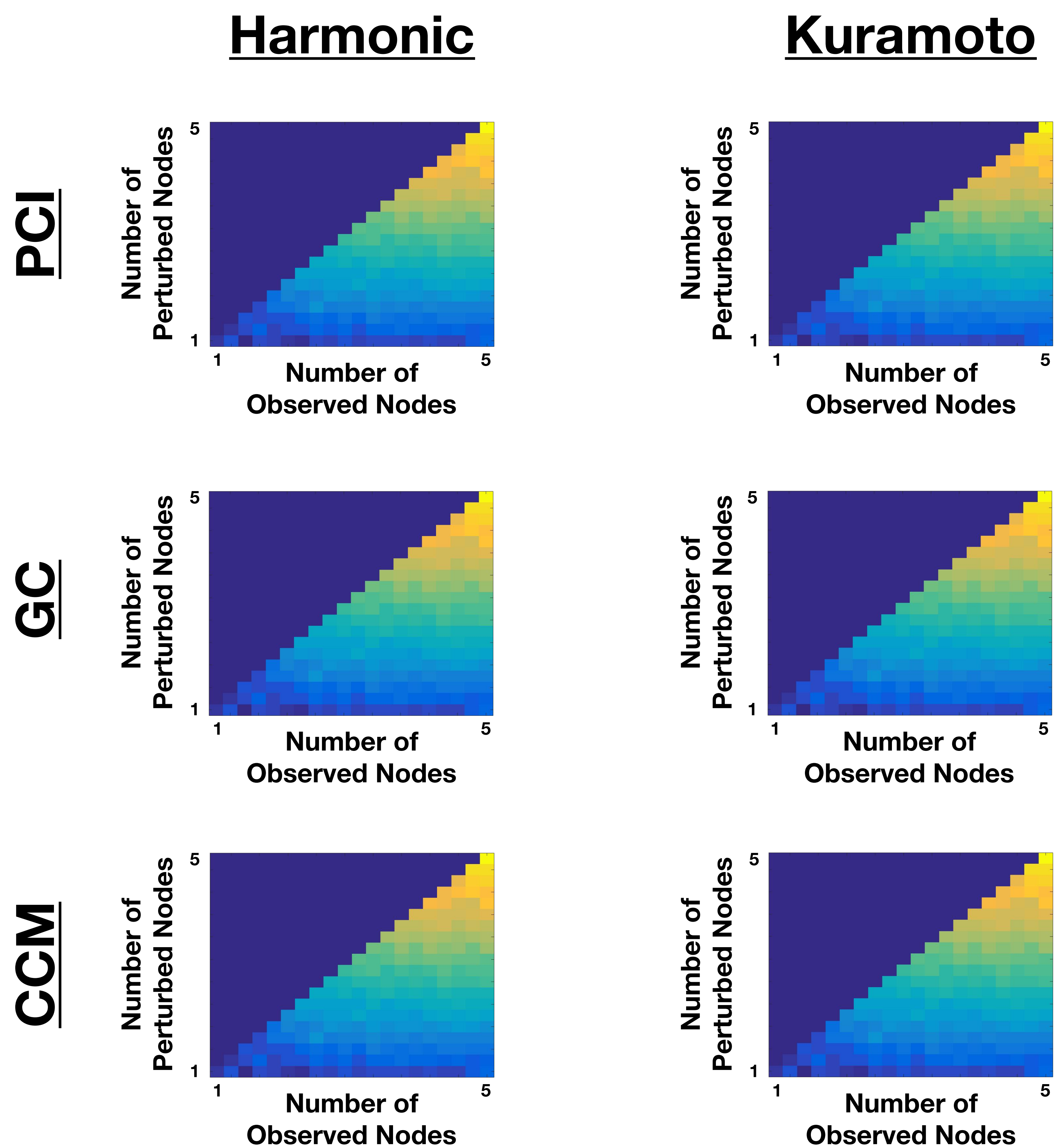
Kuramoto

True \ Pred	○ ○	○←○	○→○	○↔○
○ ○	100	0	0	0
○←○	0	100	0	0
○→○	0	0	100	0
○↔○	0	0	0	100

This figure shows that Perturbation Inference and GC do quite well (almost perfectly) on perturbed harmonic and Kuramoto oscillator data from the smallest possible network (two nodes). After this, reader might expect that perturbation inference and GC might do well on larger networks and that they do better with larger numbers of perturbations. The reader will expect that CCM will perform very poorly on perturbation data from larger networks.

Figure 10 (Part 1)

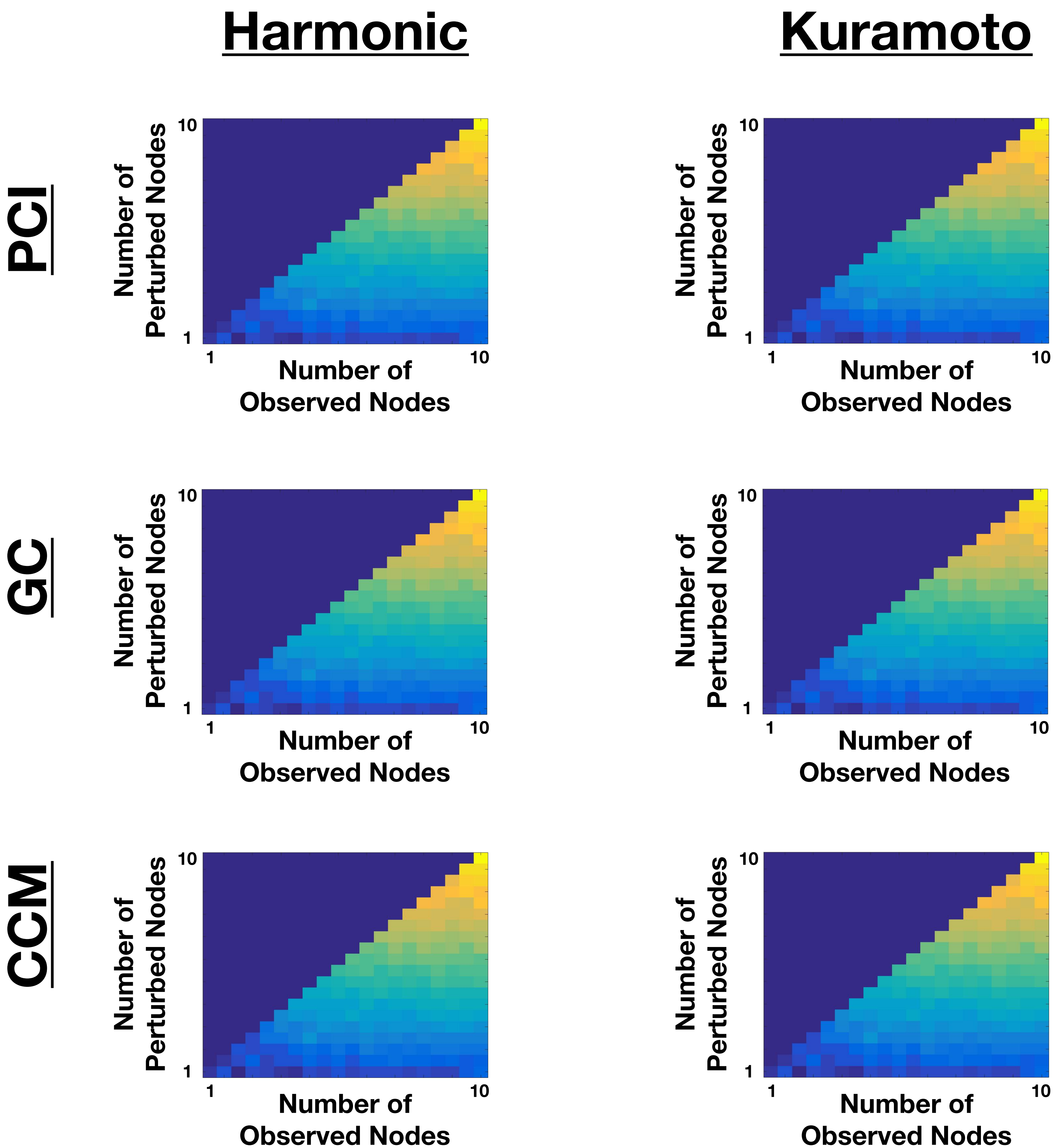
Accuracy of PCI, GC and CCM on Perturbed Data of 5 Node Harmonic and Kuramoto Oscillator Networks



This figure will show the accuracy of network inference (on 5 node networks) for PCI (perturbation causal inference), GC, and CCM over varying numbers of perturbed and observed nodes. Nodes to be perturbed are chosen randomly.

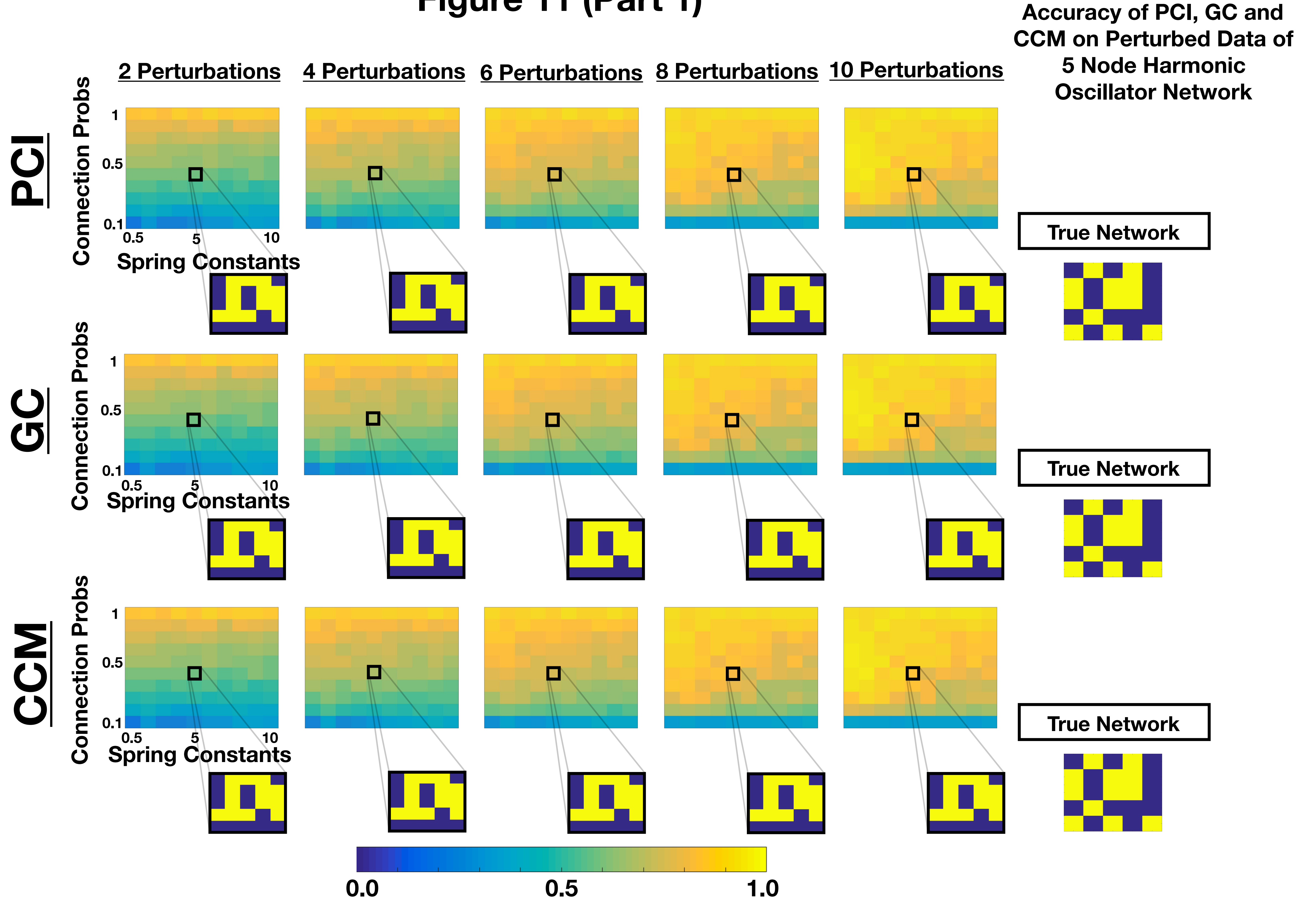
Figure 10 (Part 2)

Accuracy of PCI, GC and CCM on Perturbed Data of 10 Node Harmonic and Kuramoto Oscillator



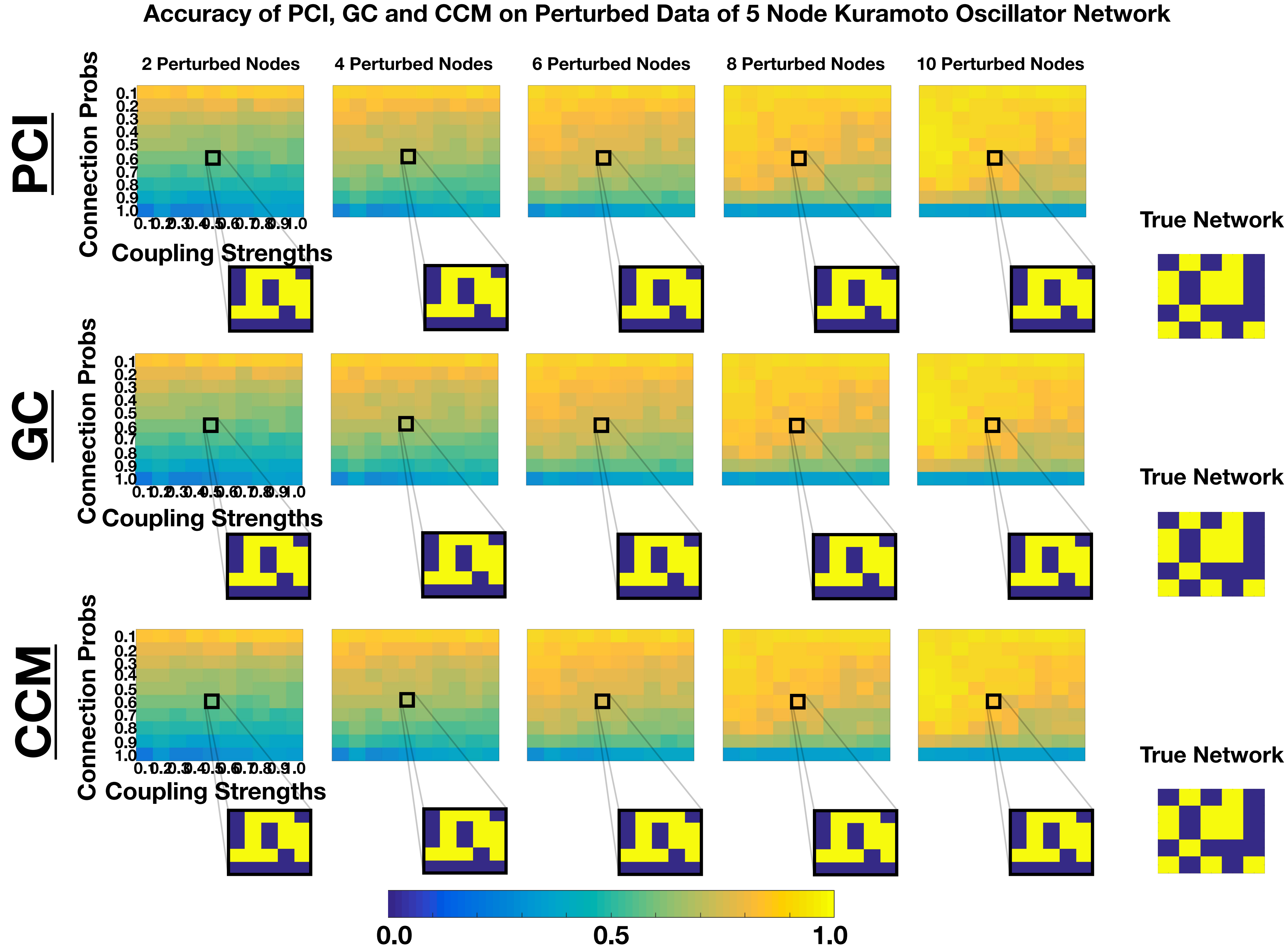
This figure will show the accuracy of network inference (on 10 node networks) for PCI (perturbation causal inference), GC, and CCM over varying numbers of perturbed and observed nodes. Nodes to be perturbed are chosen randomly.

Figure 11 (Part 1)



The graphs are all the same right now but this figure will show that PCI outperforms GC and CCM on perturbed data from 5 node harmonic oscillator networks with varying spring constants and connection probabilities. It will also be surprising because GC does quite well for 10 node harmonic oscillator networks. This will lead into my next point that GC actually does perform well in certain cases because it is a correlation based method and PCI also uses correlations for inference. This image should convince the reader that we should investigate PCI and GC further with perturbed harmonic data. CCM performs poorly so we stop investigating it after this image.

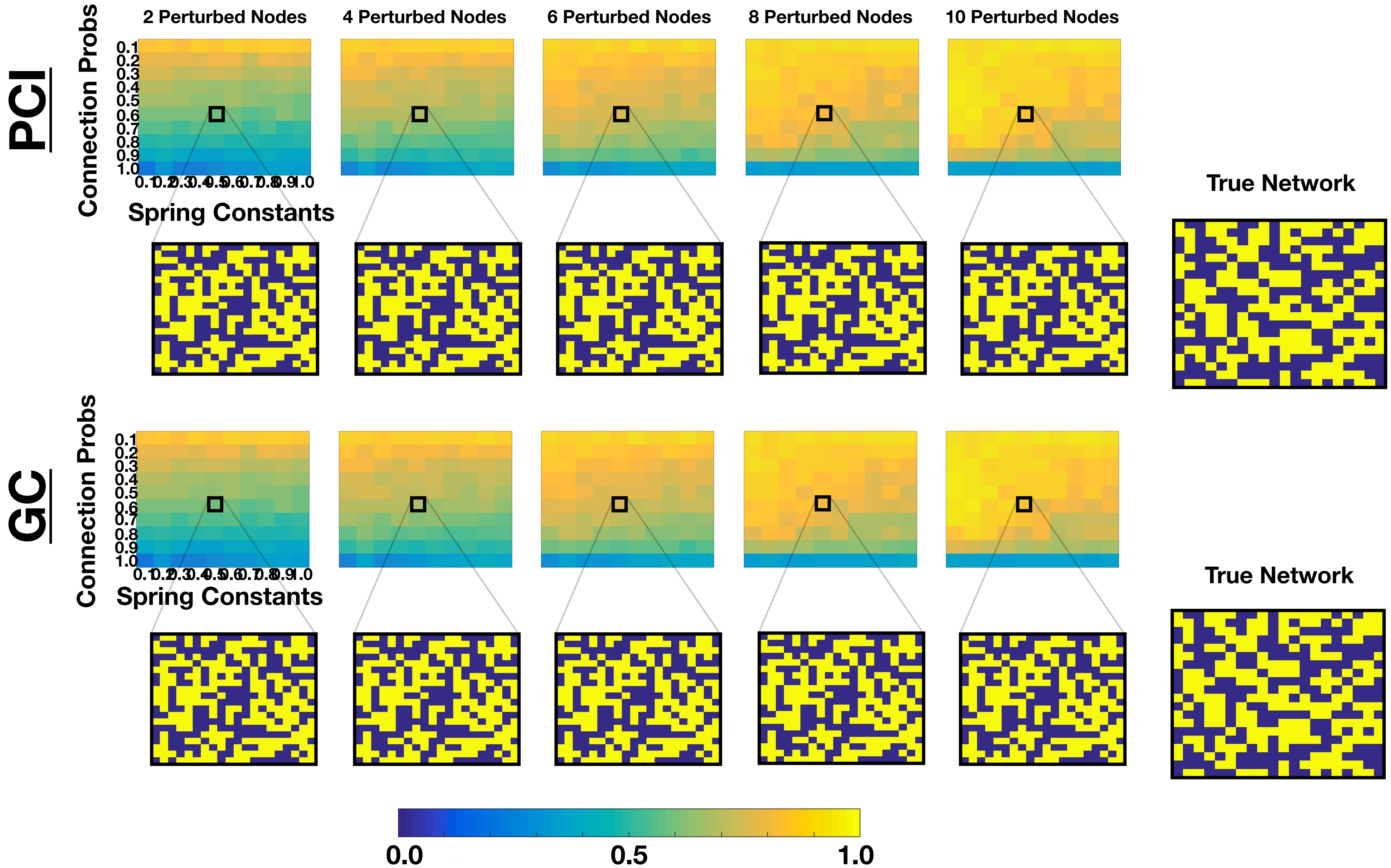
Figure 11 (Part 2)



The graphs are all the same right now but this figure will show that PCI outperforms GC and CCM on perturbed data from 5 node Kuramoto oscillator networks with varying coupling strengths and connection probabilities. Most likely, GC will not perform well on the perturbed Kuramoto data. This image should convince the reader that we should investigate PCI further with perturbed Kuramoto data. GC and CCM performs poorly on Kuramoto so we stop investigating it after this image.

Figure 12 (Part 1)

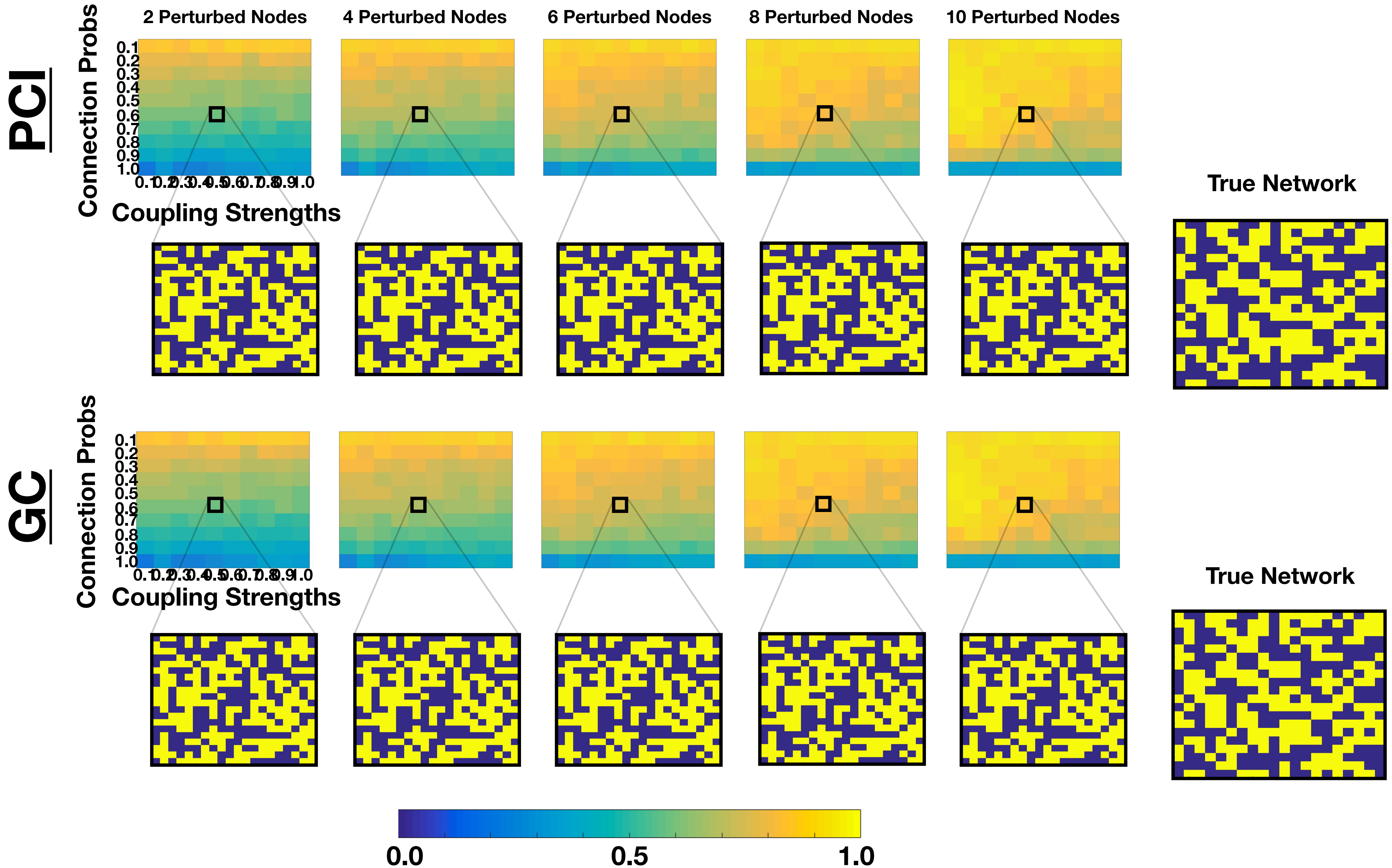
Accuracy of PCI, GC and CCM on Perturbed Data of 20 Node Harmonic Oscillator Network



The graphs are all the same right now but this figure will show that PCI and GC can scale their performance to perturbation data from 20 node harmonic oscillator networks. This image should convince the reader that PCI and GC perturbation inference is performing quite well and is scalable.

Figure 12 (Part 2)

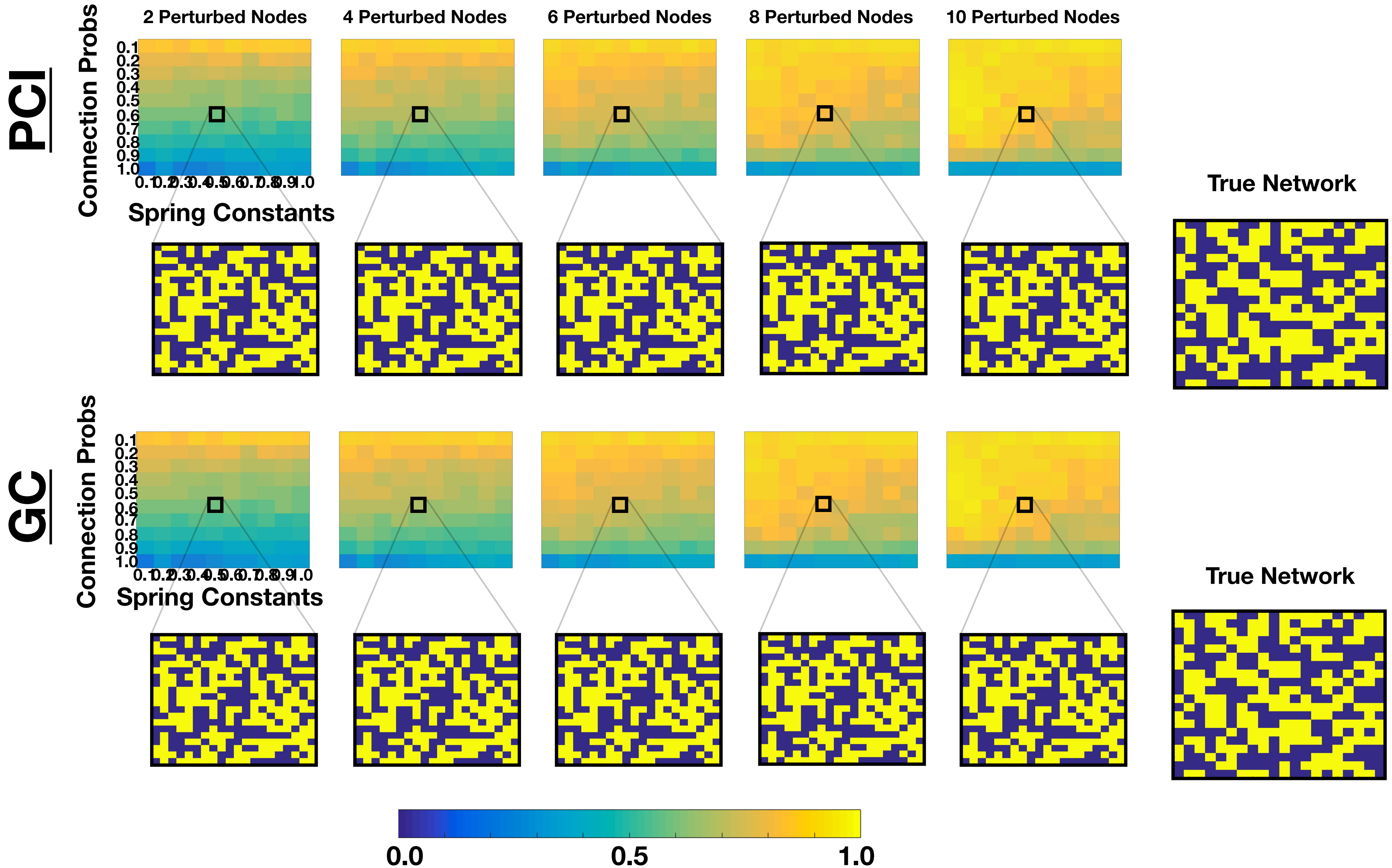
Accuracy of PCI, GC and CCM on Perturbed Data of 20 Node Kuramoto Oscillator Network



The graphs are all the same right now but this figure will show that PCI can scale its performance to perturbation data from 20 node Kuramoto oscillator networks while GC does poorly. This image should convince the reader that PCI perturbation inference is performing quite well and is scalable while GC perturbation inference works on specific systems (since it is expected to fail on the Kuramoto data).

Figure 13 (Part 1)

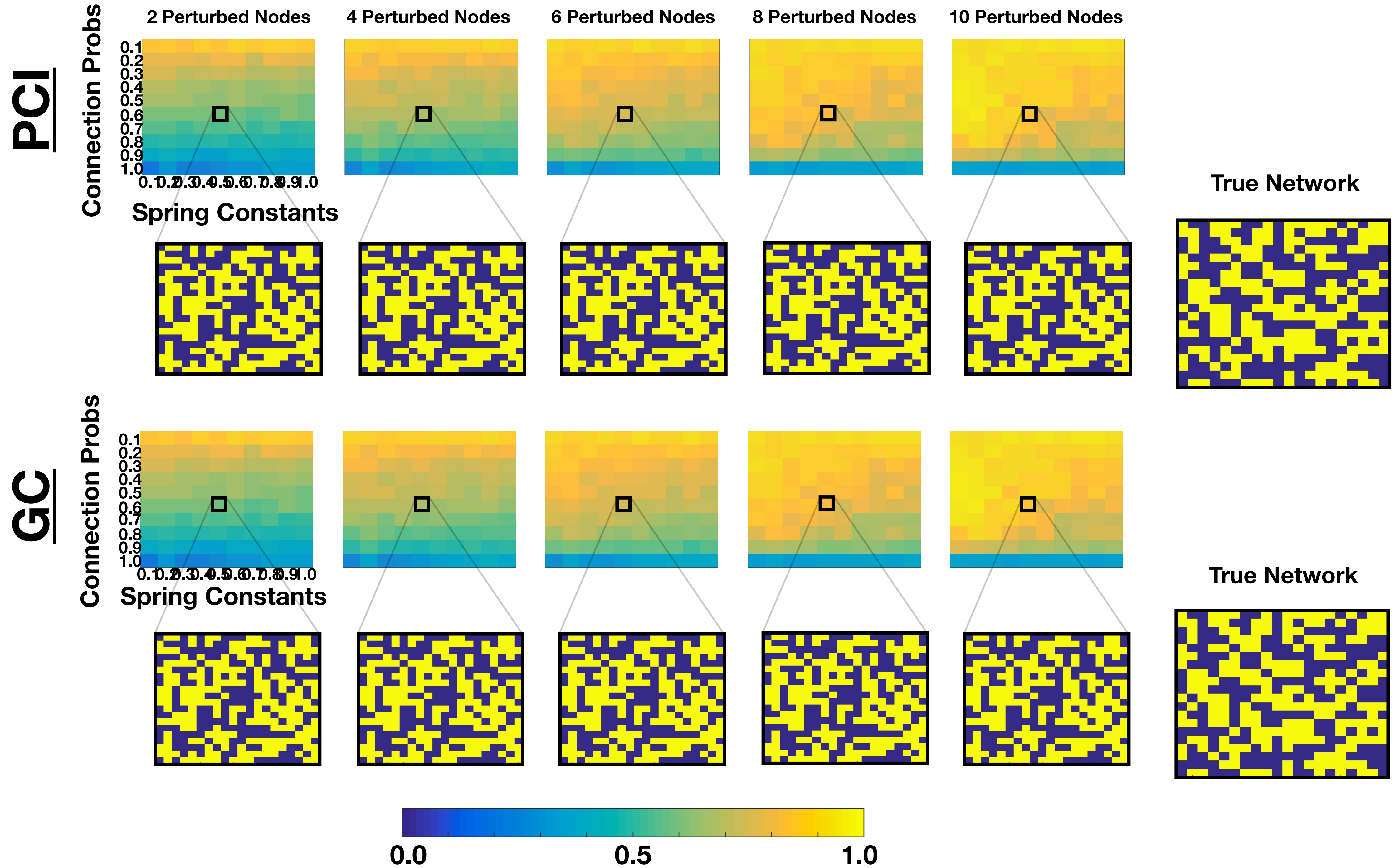
Accuracy of PCI and GC on Perturbed Data of 20 Node Harmonic Oscillator Network with Online Learning Algorithm that Chooses which Nodes to Perturb



This figure is the same as figure 12 but instead of perturbing nodes randomly, we now employ an online learning algorithm to choose the next node to perturb that will give us the most "information" about our network structure. We hope that the results with the online learning algorithm will be an improvement over the random perturbation results in figure 12. If this is true, then we should see that the predicted network looks similar to the true underlying network after fewer perturbations. This image will help the reader compare the two perturbation approaches and understand that "online learning" performs better/worse.

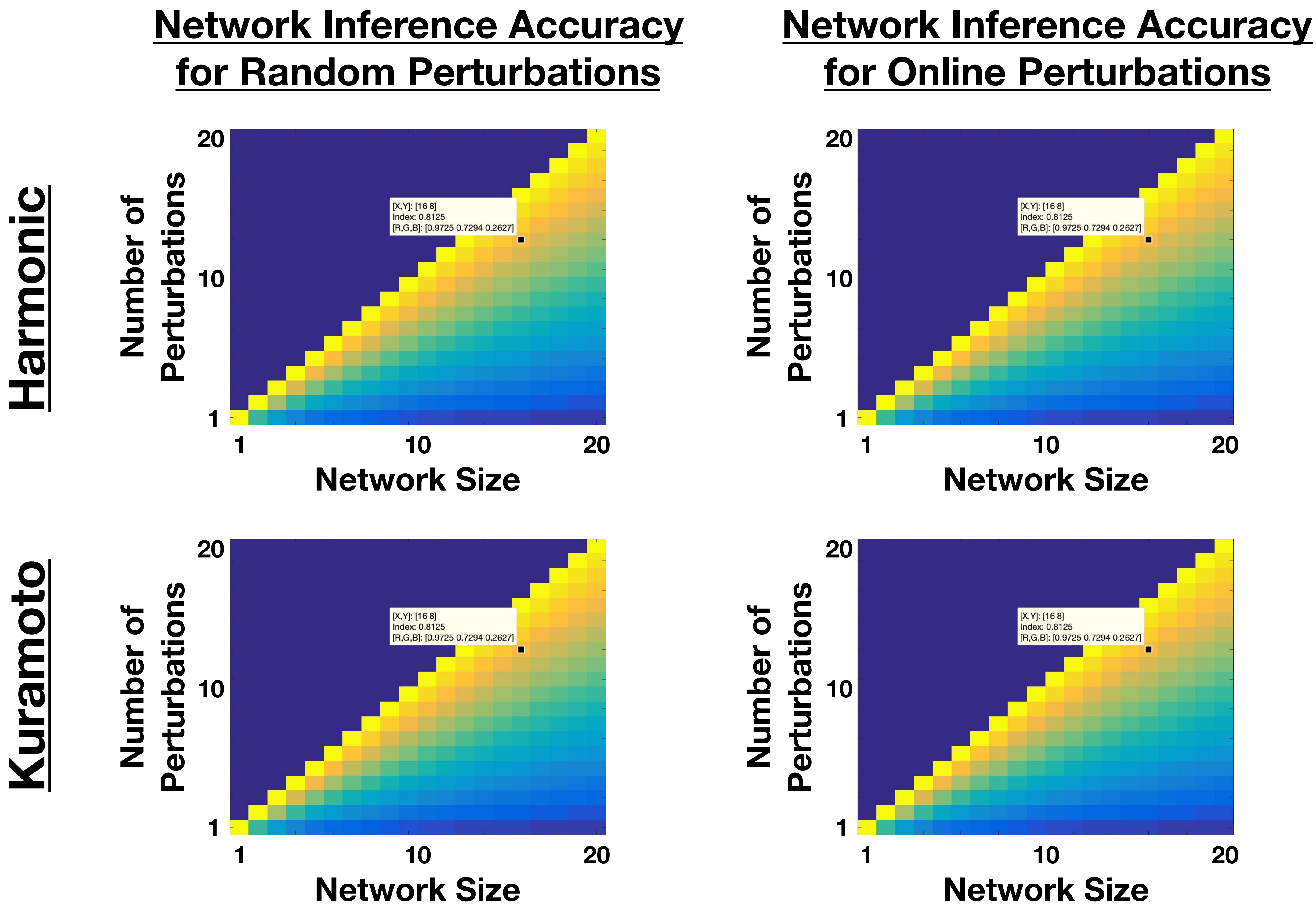
Figure 13 (Part 2)

Accuracy of PCI and GC on Perturbed Data of 20 Node Kuramoto Oscillator Network with Online Learning Algorithm that Chooses which Nodes to Perturb



This figure is the same as figure 13 but is inferring Kuramoto oscillator networks.

Figure 14



This figure will show if our online learning method outperforms choosing random nodes to perturb. If this image shows a drastic improvement using online learning, then the prospect of finding better perturbation algorithms is very exciting and may become part of the future work section in the paper. This means that with relatively few interventions, we can learn a great deal about the structure of our system.