WEBVTT

1 00:00:00.000 --> 00:00:00.990 <v Instructor>Good afternoon.</v>

2 00:00:00.990 --> 00:00:04.440 In respect for everybody's time today,

3 00:00:04.440 --> 00:00:06.570 let's go ahead and get started.

4 00:00:06.570 --> 00:00:09.300 So today, it is my pleasure to introduce

5 00:00:09.300 --> 00:00:11.550 Dr. Alexander Strang.

6 00:00:11.550 --> 00:00:15.990 Dr. Strang earned his bachelor's in mathematics, in physics,

7 00:00:15.990 --> 00:00:18.840 as well as his PhD in applied mathematics

8 00:00:18.840 --> 00:00:22.143 from Case Western Reserve University in Cleveland, Ohio.

9 00:00:23.820 --> 00:00:26.610 Born in Ohio, so representing.

10 00:00:26.610 --> 00:00:28.950 He studies variational inference problems,

11 00:00:28.950 --> 00:00:31.740 noise propagation in biological networks,

12 00:00:31.740 --> 00:00:33.810 self organizing edge flows,

13 00:00:33.810 --> 00:00:35.730 and functional form game theory

14 00:00:35.730 --> 00:00:37.710 at the University of Chicago,

15 00:00:37.710 --> 00:00:40.290 where he is a William H. Kruskal Instructor

16 00:00:40.290 --> 00:00:43.470 of physics and applied mathematics.

17 00:00:43.470 --> 00:00:46.680 Today, he's going to talk to us about motivic expansion

18 00:00:46.680 --> 00:00:50.100 of global information flow in spike train data.

19 00:00:50.100 --> 00:00:51.400 Let's welcome our speaker.

20 00:00:54.360 --> 00:00:55.980 <v ->Okay, thank you very much.</v>

21 00:00:55.980 --> 00:00:58.650 Thank you, first, for the kind invite,

22 00:00:58.650 --> 00:01:01.350 and for the opportunity to speak here in your seminar.

23 00:01:03.090 --> 00:01:06.330 So, I'd like to start with some acknowledgements.

24 00:01:06.330 --> 00:01:08.730 This is very much work in progress.

25 00:01:08.730 --> 00:01:10.800 Part of what I'm going to be showing you today

26 00:01:10.800 --> 00:01:12.390 is really the work of a Master's student

27 00:01:12.390 --> 00:01:14.670 that I've been working with this summer, that's Bowen,

28 00:01:14.670 --> 00:01:16.170 and really, I'd like to thank Bowen

29 00:01:16.170 --> 00:01:17.640 for a lot of the simulation,

30 00:01:17.640 --> 00:01:20.580 and a lot of the TE calculation I'll show you later.

31 00:01:20.580 --> 00:01:23.370 This project, more generally, was born out of conversations

32 00:01:23.370 --> 00:01:27.690 with Brent Doiron and Lek-Heng Lim here at Chicago.

33 00:01:27.690 --> 00:01:29.130 Brent really was the inspiration

34 00:01:29.130 --> 00:01:32.610 for starting to venture into computational neuroscience.

35 00:01:32.610 --> 00:01:35.253 I really say that I am new to this world,

36 00:01:35.253 --> 00:01:37.530 this world is exciting to me, but really it's a world

37 00:01:37.530 --> 00:01:41.700 that I am actively exploring and learning about.

38 00:01:41.700 --> 00:01:44.400 So I look forward to conversations afterwards

39 00:01:44.400 --> 00:01:46.170 to learn more here.

40 00:01:46.170 --> 00:01:47.940 My background was much more inspired

41 00:01:47.940 --> 00:01:50.973 by Lek-Heng's work in computational technology,

42 00:01:52.380 --> 00:01:54.300 and some of what I'll be presenting today

43 00:01:54.300 --> 00:01:56.553 is really inspired by conversations with him.

44 00:01:57.690 --> 00:02:01.200 So, let's start with some introduction and motivation.

45 00:02:01.200 --> 00:02:03.300 The motivation, personally, for this talk.

46 00:02:04.620 --> 00:02:06.420 So it goes back, really, to work that I started

47 00:02:06.420 --> 00:02:07.800 when I was a graduate student.

48 00:02:07.800 --> 00:02:10.530 I've had this long standing interest in the interplay

49 00:02:10.530 --> 00:02:14.430 between structure and dynamics, in particular in networks.

50 00:02:14.430 --> 00:02:15.570 I've been interested in questions like

51 00:02:15.570 --> 00:02:17.310 how does the structure of a network determine

52 00:02:17.310 --> 00:02:20.880 dynamics of processes on that network.

53 00:02:20.880 --> 00:02:23.700 And, in turn, how do processes on that network

54 00:02:23.700 --> 00:02:26.250 give rise to structure?

55 00:02:26.250 --> 00:02:27.830 On the biological side...

56 00:02:29.580 --> 00:02:32.370 On the biological side, in today's talk,

57 00:02:32.370 --> 00:02:36.330 I'm going to be focusing on applications of this question

58 00:02:36.330 --> 00:02:37.680 within neural networks.

59 00:02:37.680 --> 00:02:40.020 And I think that this world of computational neuroscience

60 00:02:40.020 --> 00:02:42.150 is really exciting if you're interested in this interplay

61 00:02:42.150 --> 00:02:43.920 between structure and dynamics,

62 00:02:43.920 --> 00:02:46.530 because neural networks encode, transmit, and process

63 00:02:46.530 --> 00:02:49.140 information via dynamical processes.

64 00:02:49.140 --> 00:02:53.340 For example, the process, the dynamical process

65 00:02:53.340 --> 00:02:56.160 of a neural network is directed by the wiring patterns,

66 00:02:56.160 --> 00:02:58.440 by the structure of that network, and moreover,

67 00:02:58.440 --> 00:03:00.840 if you're talking about some sort of learning process,

68 00:03:00.840 --> 00:03:03.660 then those wiring patterns can change and adapt

69 00:03:03.660 --> 00:03:06.660 during the learning process, so they are themselves dynamic.

70 00:03:07.800 --> 00:03:09.810 In this area, I've been interested in questions like,

71 00:03:09.810 --> 00:03:11.760 how is the flow of information governed

72 00:03:11.760 --> 00:03:13.500 by the wiring pattern,

73 00:03:13.500 --> 00:03:16.920 how do patterns of information flow present themselves

74 00:03:16.920 --> 00:03:19.140 in data, and can they be inferred from that data,

75 00:03:19.140 --> 00:03:20.730 and what types of wiring patterns

76 00:03:20.730 --> 00:03:22.323 might develop during learning.

77 00:03:23.910 --> 00:03:25.020 Answering questions of this type

78 00:03:25.020 --> 00:03:26.340 requires a couple of things.

79 00:03:26.340 --> 00:03:28.860 So the very, very big picture requires a language

80 00:03:28.860 --> 00:03:30.930 for describing structures and patterns,

81 00:03:30.930 --> 00:03:32.550 it requires having a dynamical process,

82 00:03:32.550 --> 00:03:35.040 some sort of model for the neural net,

83 00:03:35.040 --> 00:03:37.530 and it requires a generating model

84 00:03:37.530 --> 00:03:40.080 that generates initial structure,

85 00:03:40.080 --> 00:03:42.330 and links structure to dynamics.

86 00:03:42.330 --> 00:03:45.420 Alternatively, if we don't generate things using a model,

87 00:03:45.420 --> 00:03:47.460 if we have some sort of observable or data,

88 00:03:47.460 --> 00:03:49.020 then we can try to work in the other direction

89 00:03:49.020 --> 00:03:51.540 and go from dynamics to structure.

90 00:03:51.540 --> 00:03:54.150 Today, during this talk, I'm going to be focusing really

91 00:03:54.150 --> 00:03:55.320 on this first piece,

92 00:03:55.320 --> 00:03:57.480 on a language for describing structures and patterns,

93 00:03:57.480 --> 00:03:58.560 and on the second piece,

94 00:03:58.560 --> 00:04:01.350 on an observable that I've been working on

95 00:04:01.350 --> 00:04:05.010 trying to compute to use to try to connect

96 00:04:05.010 --> 00:04:07.530 these three points together.

97 00:04:07.530 --> 00:04:10.140 So, to get started, a little bit of biology.

98 00:04:10.140 --> 00:04:12.540 Really, I was inspired in this project by a paper

99 00:04:12.540 --> 00:04:14.485 from Keiji Miura.

100 00:04:14.485 --> 00:04:16.650 He was looking at a coupled oscillator model,

101 00:04:16.650 --> 00:04:19.770 this was a Kuramoto model for neural activity

102 00:04:19.770 --> 00:04:22.140 where the firing dynamics interact with the wiring.

103 00:04:22.140 --> 00:04:25.650 So the wiring that couples the oscillators

104 00:04:25.650 --> 00:04:28.860 would adapt on a slower timescale

105 00:04:28.860 --> 00:04:31.440 than the oscillators themselves,

106 00:04:31.440 --> 00:04:33.570 and that adaptation could represent

107 00:04:33.570 --> 00:04:35.970 different types of learning processes.

108 00:04:35.970 --> 00:04:39.660 For example, the fire-together wire-together rules,

109 00:04:39.660 --> 00:04:40.560 so Hebbian learning,

110 00:04:40.560 --> 00:04:43.110 you could look at causal learning rules,

111 00:04:43.110 --> 00:04:44.790 or anti-Hebbian learning rules.

112 00:04:44.790 --> 00:04:48.240 This is just an example of one, of the system.

113 00:04:48.240 --> 00:04:49.980 This system of (indistinct) is sort of interesting

114 00:04:49.980 --> 00:04:52.410 because it can generate all sorts of different patterns.

115 00:04:52.410 --> 00:04:53.910 You can see synchronized firing,

116 00:04:53.910 --> 00:04:55.110 you can see traveling waves,

117 00:04:55.110 --> 00:04:56.610 you can see chaos.

118 00:04:56.610 --> 00:04:59.280 These occur at different critical boundaries.

119 00:04:59.280 --> 00:05:01.170 So you can see phase transmissions

120 00:05:01.170 --> 00:05:03.570 when you have large collections of these oscillators.

121 00:05:03.570 --> 00:05:05.100 And depending on how they're coupled together,

122 00:05:05.100 --> 00:05:06.333 it behaves differently.

123 00:05:07.410 --> 00:05:10.320 In particular, what's interesting here is that

124 00:05:10.320 --> 00:05:13.350 starting from some random seed topology,

125 00:05:13.350 --> 00:05:16.170 the dynamics play forward from that initial condition,

126 00:05:16.170 --> 00:05:17.910 and that random seed topology

127 00:05:17.910 --> 00:05:19.920 produces an ensemble of wiring patterns

128 00:05:19.920 --> 00:05:21.527 that are themselves random.

129 00:05:21.527 --> 00:05:23.850 And we can think of that ensemble of wiring patterns

130 00:05:23.850 --> 00:05:28.083 as being chaotic realizations of some random initialization.

131 00:05:29.460 --> 00:05:31.560 That said, you can also observe structures

132 00:05:31.560 --> 00:05:33.360 within the systems of coupled oscillators.

133 00:05:33.360 --> 00:05:35.670 So you can see large scale cyclic structures

134 00:05:35.670 --> 00:05:37.830 representing organized causal firing patterns

135 00:05:37.830 --> 00:05:39.840 in certain regimes.

136 00:05:39.840 --> 00:05:40.980 So this is a nice example

137 00:05:40.980 --> 00:05:42.510 where graph structure and learning parameters

138 00:05:42.510 --> 00:05:44.460 can determine dynamics, and in turn,

139 00:05:44.460 --> 00:05:46.710 where those dynamics can determine structure.

140 00:05:48.030 --> 00:05:49.440 On the other side, you can also think about

141 00:05:49.440 --> 00:05:51.940 a data-driven side instead of a model-driven side.

142 00:05:53.460 --> 00:05:55.590 If we empirically observe sample trajectories

143 00:05:55.590 --> 00:05:57.720 of some observables, for example, neuron recordings,

144 00:05:57.720 --> 00:05:59.070 then we might hope to infer something

145 00:05:59.070 --> 00:06:01.050 about the connectivity that generates them.

146 00:06:01.050 --> 00:06:03.750 And so here, instead of starting by posing a model,

147 00:06:03.750 --> 00:06:06.000 and then simulating it and setting up how it behaves,

148 00:06:06.000 --> 00:06:07.440 we can instead study data,

149 00:06:07.440 --> 00:06:09.900 or try to study structure in data.

150 00:06:09.900 --> 00:06:12.420 Often, that data comes in the form of covariance matrices

151 00:06:12.420 --> 00:06:14.040 representing firing rates.

152 00:06:14.040 --> 00:06:15.330 And these covariance matrices

153 00:06:15.330 --> 00:06:19.110 may be auto covariance matrices with some sort of time-lag.

154 00:06:19.110 --> 00:06:21.660 Here, there are a couple of standard structural approaches.

155 00:06:21.660 --> 00:06:24.540 So there's a motivic expansion approach.

156 00:06:24.540 --> 00:06:28.350 This was at least introduced by Brent Doiron's lab,

157 00:06:28.350 --> 00:06:30.450 with his student, Gabe Ocker.

158 00:06:30.450 --> 00:06:33.600 Here, the idea is that you define some graph motifs,

159 00:06:33.600 --> 00:06:35.730 and then you can study the dynamics

160 00:06:35.730 --> 00:06:37.530 in terms of those graph motifs.

161 00:06:37.530 --> 00:06:41.010 For example, if you build a power series in those motifs,

162 00:06:41.010 --> 00:06:43.770 then you can try to represent your covariance matrices

163 00:06:43.770 --> 00:06:45.060 in terms of that power series.

164 00:06:45.060 --> 00:06:46.170 And this is something we're gonna talk about

165 00:06:46.170 --> 00:06:47.130 quite a bit today.

166 00:06:47.130 --> 00:06:49.350 This really, part of why I was inspired by this work is

167 00:06:49.350 --> 00:06:50.670 I had been working separately

168 00:06:50.670 --> 00:06:52.650 on the idea of looking at covariance matrices

169 00:06:52.650 --> 00:06:54.903 in terms of these power series expansions.

170 00:06:56.040 --> 00:06:59.160 This is also connected to topological data analysis,

171 00:06:59.160 --> 00:07:01.170 and this is where the conversations with Lek-Heng

172 00:07:01.170 --> 00:07:02.940 played a role in this work.

173 00:07:02.940 --> 00:07:06.690 Topological data analysis aims to construct graphs

174 00:07:06.690 --> 00:07:08.460 representing dynamical systems.

175 00:07:08.460 --> 00:07:10.538 For example, you might look at the dynamical similarity

176 00:07:10.538 --> 00:07:12.990 of firing patterns of certain neurons,

177 00:07:12.990 --> 00:07:16.743 and then try to study the topology of those graphs.

178 00:07:17.730 --> 00:07:19.530 Again, this leads to similar questions,

179 00:07:19.530 --> 00:07:21.120 but we could be a little bit more precise here

180 00:07:21.120 --> 00:07:22.570 for thinking in neuroscience.

181 00:07:23.580 --> 00:07:25.350 We can say more precisely, for example,

182 00:07:25.350 --> 00:07:28.590 how is information processing and transfer represented,

183 00:07:28.590 --> 00:07:30.570 both in these covariance matrices

184 00:07:30.570 --> 00:07:33.390 and the structures that we hope to extract from them?

185 00:07:33.390 --> 00:07:36.330 In particular, can we try and infer causality

186 00:07:36.330 --> 00:07:37.893 from firing patterns?

187 00:07:39.420 --> 00:07:41.847 And this is fundamentally an information theoretic question.

188 00:07:41.847 --> 00:07:42.870 And really, we're asking,

189 00:07:42.870 --> 00:07:45.420 can we study the directed exchange of information

190 00:07:45.420 --> 00:07:47.400 from trajectories?

191 00:07:47.400 --> 00:07:49.320 Here, one approach, I mean, in some sense,

192 00:07:49.320 --> 00:07:52.740 you can never tell causality without some underlying model,

193 00:07:52.740 --> 00:07:55.770 without some underlying understanding and mechanism,

194 00:07:55.770 --> 00:07:57.540 so if all we can do is observe,

195 00:07:57.540 --> 00:08:00.510 then we need to define what we mean by causality.

196 00:08:00.510 --> 00:08:03.780 A reasonable standard definition here is Wiener causality,

197 00:08:03.780 --> 00:08:06.180 which says that two time series share a causal relation,

198 00:08:06.180 --> 00:08:08.040 so we say x causes y,

199 00:08:08.040 --> 00:08:11.520 if the history of x informs the future of y.

200 00:08:11.520 --> 00:08:14.250 And note that here, cause, I put in quotes,

201 00:08:14.250 --> 00:08:15.540 really means forecasts.

202 00:08:15.540 --> 00:08:18.180 It means that the past, or the present of x,

203 00:08:18.180 --> 00:08:21.630 carries relevant information about the future of y.

204 00:08:21.630 --> 00:08:26.190 A natural measure of that information is transfer entropy.

205 00:08:26.190 --> 00:08:29.715 Transfer entropy was introduced by Schrieber in 2000,

206 00:08:29.715 --> 00:08:31.530 and is the expected KL divergence

207 00:08:31.530 --> 00:08:35.340 between the distribution of the future of y

208 00:08:35.340 --> 00:08:38.010 given the history of x,

209 00:08:38.010 --> 00:08:41.130 and the marginal distribution of the future of y.

210 00:08:41.130 --> 00:08:43.110 So essentially, it's how much predictive information

211 00:08:43.110 --> 00:08:44.763 does x carry about y.

212 00:08:46.080 --> 00:08:48.450 This is a nice quantity for a couple of reasons.

213 00:08:48.450 --> 00:08:51.330 First, it's zero when two trajectories are independent.

214 00:08:51.330 --> 00:08:52.920 Second, since it's just defining

215 00:08:52.920 --> 00:08:55.500 some of these conditional distributions, it's model free,

216 00:08:55.500 --> 00:08:57.510 so I put here no with a star,

217 00:08:57.510 --> 00:09:00.660 because generative assumptions actually do matter

218 00:09:00.660 --> 00:09:02.340 when you go to try and compute it, but in principle,

219 00:09:02.340 --> 00:09:04.530 it's defined independent of the model.

220 00:09:04.530 --> 00:09:07.470 Again, unlike some other effective causality measures,

221 00:09:07.470 --> 00:09:11.340 it doesn't require introducing some time-lag to define.

222 00:09:11.340 --> 00:09:13.350 It's a naturally directed quantity.

223 00:09:13.350 --> 00:09:14.640 We can say that the future of y

224 00:09:14.640 --> 00:09:16.680 conditioned on the past of x...

225 00:09:16.680 --> 00:09:20.370 That transfer entropy is defined in terms of the future of y

226 00:09:20.370 --> 00:09:22.830 conditioned on the past of x and y.

227 00:09:22.830 --> 00:09:27.090 And that quantity is directed, because reversing x and y

228 00:09:27.090 --> 00:09:29.670 does not symmetrically change the statement.

229 00:09:29.670 --> 00:09:31.860 This is different than quantities like mutual information

230 00:09:31.860 --> 00:09:34.290 or correlation, that are also often used

231 00:09:34.290 --> 00:09:36.870 to try and measure effective connectivity in networks

232 00:09:36.870 --> 00:09:39.843 which are fundamentally symmetric quantities.

233 00:09:41.400 --> 00:09:42.960 Transfer entropy is also less corrupted

234 00:09:42.960 --> 00:09:45.840 by measurement noise, linear mixing of signals,

235 00:09:45.840 --> 00:09:48.393 or shared coupling to external sources.

236 00:09:49.800 --> 00:09:51.870 Lastly, and maybe most interestingly,

237 00:09:51.870 --> 00:09:54.000 if we think in terms of correlations,

238 00:09:54.000 --> 00:09:55.590 correlations are really moments,

239 00:09:55.590 --> 00:09:57.360 correlations are really about covariances, right,

240 00:09:57.360 --> 00:09:58.980 second order moments.

241 00:09:58.980 --> 00:10:00.810 Transfer entropies, these are about entropies,

242 00:10:00.810 --> 00:10:03.780 these are logs of distributions,

243 00:10:03.780 --> 00:10:06.360 and so they depend on the full shape of these distributions.

244 00:10:06.360 --> 00:10:09.870 Which means that transfer entropy can capture coupling

245 00:10:09.870 --> 00:10:13.080 that is maybe not apparent, or not obvious

246 00:10:13.080 --> 00:10:16.203 just looking at second order moment type analysis.

247 00:10:17.280 --> 00:10:20.070 So transfer entropy has been applied pretty broadly.

248 00:10:20.070 --> 00:10:22.440 It's been applied to spiking cortical networks

249 00:10:22.440 --> 00:10:23.610 and calcium imaging,

250 00:10:23.610 --> 00:10:28.560 to MEG data in motor tasks and auditory discrimination,

251 00:10:28.560 --> 00:10:30.570 it's been applied to emotion recognition,

252 00:10:30.570 --> 00:10:31.740 precious metal prices

253 00:10:31.740 --> 00:10:34.050 and multivariate time series forecasting,

254 00:10:34.050 --> 00:10:36.180 and more recently, to accelerate learning

255 00:10:36.180 --> 00:10:38.040 in different artificial neural nets.

256 00:10:38.040 --> 00:10:39.990 So you can look at feedforward architectures,

257 00:10:39.990 --> 00:10:42.450 convolutional architectures, even recurrent neural nets.

258 00:10:42.450 --> 00:10:45.120 And transfer entropy has been used to accelerate learning

259 00:10:45.120 --> 00:10:46.443 in those frameworks.

260 00:10:48.570 --> 00:10:49.590 For this part of the talk,

261 00:10:49.590 --> 00:10:52.470 I'd like to focus really on two questions.

262 00:10:52.470 --> 00:10:55.050 First, how do we compute transfer entropy,

263 00:10:55.050 --> 00:10:58.380 and then second, if we could compute transfer entropy,

264 00:10:58.380 --> 00:10:59.700 and build a graph out of that,

265 00:10:59.700 --> 00:11:01.410 how would we study the structure of that graph?

266 00:11:01.410 --> 00:11:04.053 Essentially, how is information flow structured?

267 00:11:05.460 --> 00:11:07.660 We'll start with computing transfer entropy.

268 00:11:09.120 --> 00:11:10.140 To compute transfer entropy,

269 00:11:10.140 --> 00:11:12.540 we actually need to write down an equation.

270 00:11:12.540 --> 00:11:14.400 So transfer entropy was originally introduced

271 00:11:14.400 --> 00:11:17.820 for discrete time arbitrary order Markov processes.

272 00:11:17.820 --> 00:11:20.337 So suppose we have two Markov processes X and Y.

273 00:11:20.337 --> 00:11:24.997 And we'll let Xn denote the state of process X at time n,

274 00:11:24.997 --> 00:11:27.390 and Xnk, where the k is in superscript,

275 00:11:27.390 --> 00:11:31.170 to denote the sequence starting from n minus k plus 1

276 00:11:31.170 --> 00:11:32.010 going up to n.

277 00:11:32.010 --> 00:11:37.010 So that's the last k states that the process X visited.

278 00:11:37.260 --> 00:11:39.990 Then, the transfer entropy from Y to X,

279 00:11:39.990 --> 00:11:42.663 they're denoted T, Y over to X,

280 00:11:43.980 --> 00:11:48.980 is the entropy of the future of X, conditioned on its past,

281 00:11:50.130 --> 00:11:53.640 minus the entropy of the future of X conditioned on its past

282 00:11:53.640 --> 00:11:56.280 and the past of the trajectory Y.

283 00:11:56.280 --> 00:11:57.320 So here, you can think the transfer entropy

284 00:11:57.320 --> 00:11:58.950 is essentially the reduction in entropy

285 00:11:58.950 --> 00:12:03.450 of the future states of X when incorporating the past of Y.

286 00:12:03.450 --> 00:12:04.950 This means that computing transfer entropy

287 00:12:04.950 --> 00:12:07.140 reduces to estimating essentially these entropies.

288 00:12:07.140 --> 00:12:08.850 That means we need to estimate essentially

289 00:12:08.850 --> 00:12:12.633 the conditional distributions inside of these parentheses.

290 00:12:13.620 --> 00:12:16.410 That's easy for certain processes, so for example,

291 00:12:16.410 --> 00:12:18.660 if X and Y are Gaussian processes,

292 00:12:18.660 --> 00:12:20.160 then really what we're trying to compute

293 00:12:20.160 --> 00:12:21.690 is conditional mutual information,

294 00:12:21.690 --> 00:12:22.800 and there are nice equations

295 00:12:22.800 --> 00:12:24.510 for conditional mutual information

296 00:12:24.510 --> 00:12:26.220 when you have Gaussian random variables.

297 00:12:26.220 --> 00:12:29.250 So if I have three Gaussian random variables, X, Y, Z,

298 00:12:29.250 --> 00:12:32.700 possibly multivariate, with joint covariant sigma,

299 00:12:32.700 --> 00:12:34.560 then the conditional mutual information

300 00:12:34.560 --> 00:12:37.140 between these variables, so the mutual information

301 00:12:37.140 --> 00:12:38.910 between X and Y conditioned on Z,

302 00:12:38.910 --> 00:12:41.610 is just given by this ratio of log determinants

303 00:12:41.610 --> 00:12:42.710 of those convariances.

304 00:12:44.970 --> 00:12:48.210 In particular, a common test model used

305 00:12:48.210 --> 00:12:50.520 in the transfer entropy literature

306 00:12:50.520 --> 00:12:52.530 are linear auto-regressive processes,

307 00:12:52.530 --> 00:12:54.600 because a linear auto-regressive process,

308 00:12:54.600 --> 00:12:56.550 when perturbed by Gaussian noise,

309 00:12:56.550 --> 00:12:58.200 produces a Gaussian process.

310 00:12:58.200 --> 00:12:59.910 All of the different joint marginal

311 00:12:59.910 --> 00:13:01.770 conditional distributions are all Gaussian,

312 00:13:01.770 --> 00:13:03.090 which means that we can compute

313 00:13:03.090 --> 00:13:05.610 these covariances analytically, which then means

314 00:13:05.610 --> 00:13:07.290 that you can compute the transfer entropy analytically.

315 00:13:07.290 --> 00:13:08.940 So these linear auto-regressive processes

316 00:13:08.940 --> 00:13:10.080 are nice test cases,

317 00:13:10.080 --> 00:13:12.450 'cause you can do everything analytically.

318 00:13:12.450 --> 00:13:14.880 They're also somewhat disappointing, or somewhat limiting,

319 00:13:14.880 --> 00:13:17.340 because in this linear auto-regressive case,

320 00:13:17.340 --> 00:13:20.223 transfer entropy is the same as Granger causality.

321 00:13:21.630 --> 00:13:24.780 And in this Gaussian case, essentially what we've done

322 00:13:24.780 --> 00:13:26.610 is we've reduced transfer entropy

323 00:13:26.610 --> 00:13:28.530 to a study of time-lagged correlations.

324 00:13:28.530 --> 00:13:31.530 So this becomes the same as a correlation based analysis.

325 00:13:31.530 --> 00:13:34.350 We can't incorporate information beyond the second moments

326 00:13:34.350 --> 00:13:36.390 if we restrict ourselves to Gaussian processes,

327 00:13:36.390 --> 00:13:38.520 or Gaussian approximations.

328 00:13:38.520 --> 00:13:41.130 The other thing to note is this is strongly model dependent,

329 00:13:41.130 --> 00:13:42.630 because this particular formula

330 00:13:42.630 --> 00:13:43.890 for computing mutual information

331 00:13:43.890 --> 00:13:46.383 depends on having Gaussian distributions.

332 00:13:49.530 --> 00:13:53.220 In a more general setting, or a more empirical setting,

333 00:13:53.220 --> 00:13:54.960 you might observe some data.

334 00:13:54.960 --> 00:13:56.130 You don't know if that data

335 00:13:56.130 --> 00:13:57.777 comes from some particular process,

336 00:13:57.777 --> 00:13:59.340 and you can't necessarily assume

337 00:13:59.340 --> 00:14:01.080 the conditional distribution is Gaussian.

338 00:14:01.080 --> 00:14:03.420 But we would still like to estimate transfer entropy,

339 00:14:03.420 --> 00:14:05.640 which leads to the problem of estimating transfer entropy

340 00:14:05.640 --> 00:14:08.040 given an observed time series.

341 00:14:08.040 --> 00:14:10.530 We would like to do this, again, sans model assumptions,

342 00:14:10.530 --> 00:14:13.140 so we don't want to assume Gaussianity.

343 00:14:13.140 --> 00:14:15.720 This is sort of trivial, again, I star that,

344 00:14:15.720 --> 00:14:16.920 in discrete state spaces,

345 00:14:16.920 --> 00:14:19.800 because essentially it amounts to counting occurrences.

346 00:14:19.800 --> 00:14:22.920 But it becomes difficult whenever the state spaces are large

347 00:14:22.920 --> 00:14:25.473 and/or high dimensional, as they often are.

348 00:14:26.340 --> 00:14:28.440 This leads to a couple of different approaches.

349 00:14:28.440 --> 00:14:31.890 So, as a first example, let's consider spike train data.

350 00:14:31.890 --> 00:14:34.170 So spike train data consists, essentially,

351 00:14:34.170 --> 00:14:38.700 of binning the state of a neuron into either on or off.

352 00:14:38.700 --> 00:14:41.460 So neurons, you can think either in a state zero or one.

353 00:14:41.460 --> 00:14:44.490 And then a pairwise calculation for transfer entropy

354 00:14:44.490 --> 00:14:47.640 only requires estimating a joint probability distribution

355 00:14:47.640 --> 00:14:50.910 over 4 to the k plus l states, where k plus l,

356 00:14:50.910 --> 00:14:53.970 k is the history of x that we remember,

357 00:14:53.970 --> 00:14:55.860 and l is the history of y.

358 00:14:55.860 --> 00:15:00.860 So if the Markov process generating the spike train data

359 00:15:01.350 --> 00:15:04.200 is not of high order, does not have a long memory,

360 00:15:04.200 --> 00:15:06.390 then these k and l can be small,

361 00:15:06.390 --> 00:15:08.160 and this state space is fairly small,

362 00:15:08.160 --> 00:15:09.900 so this falls into that first category,

363 00:15:09.900 --> 00:15:11.520 when we're looking at a discrete state space

364 00:15:11.520 --> 00:15:13.023 and it's not too difficult.

365 00:15:14.880 --> 00:15:17.640 In a more general setting, if we don't try to bin the states

366 00:15:17.640 --> 00:15:19.380 of the neurons to on or off,

367 00:15:19.380 --> 00:15:22.110 for example, maybe we're looking at a firing rate model,

368 00:15:22.110 --> 00:15:23.970 where we want to look at the firing rates of the neurons,

369 00:15:23.970 --> 00:15:27.210 and that's a continuous random variable,

370 00:15:27.210 --> 00:15:29.250 then we need some other types of estimators.

371 00:15:29.250 --> 00:15:30.720 So the common estimator used here

372 00:15:30.720 --> 00:15:33.600 is a kernel density estimator, or KSG estimator.

373 00:15:33.600 --> 00:15:35.790 And this is designed for large, continuous,

374 00:15:35.790 --> 00:15:37.110 or high dimensional state spaces,

375 00:15:37.110 --> 00:15:39.273 e.g., these firing rate models.

376 00:15:40.170 --> 00:15:43.320 Typically, the approach is to employ a Takens delay map,

377 00:15:43.320 --> 00:15:45.120 which embeds your high dimensional data

378 00:15:45.120 --> 00:15:47.670 in some sort of lower dimensional space,

379 00:15:47.670 --> 00:15:50.250 that tries to capture the intrinsic dimension

380 00:15:50.250 --> 00:15:54.630 of the attractor that your dynamic process settles onto.

381 00:15:54.630 --> 00:15:56.970 And then you try to estimate an unknown density

382 00:15:56.970 --> 00:15:59.730 based on this delay map using a k-nearest neighbor

383 00:15:59.730 --> 00:16:01.080 kernel density estimate.

384 00:16:01.080 --> 00:16:04.290 The advantage of this sort of k-nearest neighbor

385 00:16:04.290 --> 00:16:06.060 kernel density is it dynamically adapts

386 00:16:06.060 --> 00:16:08.640 the width of the kernel given your sample density.

387 00:16:08.640 --> 00:16:11.310 And this has been implemented in some open source toolkits.

388 00:16:11.310 --> 00:16:13.493 These are the toolkits that we've been working with.

389 00:16:15.210 --> 00:16:17.640 So we've tested this on a couple of different models.

390 00:16:17.640 --> 00:16:18.780 And really, I'd say this work,

391 00:16:18.780 --> 00:16:20.310 this is still very much work in progress,

392 00:16:20.310 --> 00:16:23.130 this is work that Bowen was developing over the summer.

393 00:16:23.130 --> 00:16:26.490 And so we developed a couple of different models to test.

394 00:16:26.490 --> 00:16:29.310 The first were these linear auto-regressive networks,

395 00:16:29.310 --> 00:16:30.210 and we just used these

396 00:16:30.210 --> 00:16:31.800 to test the accuracy of the estimators,

397 00:16:31.800 --> 00:16:34.140 because everything here is Gaussian, so you can compute

398 00:16:34.140 --> 00:16:36.900 the necessary transfer entropies analytically.

399 00:16:36.900 --> 00:16:38.820 The next, more interesting class of networks

400 00:16:38.820 --> 00:16:41.520 are threshold linear networks, or TLNs.

401 00:16:41.520 --> 00:16:44.490 These are a firing rate model, where your rate, r,

402 00:16:44.490 --> 00:16:46.590 obeys this stochastic differential equation.

403 00:16:46.590 --> 00:16:50.940 So the rate of change in the rate, dr(t), is...

404 00:16:50.940 --> 00:16:54.690 So you have sort of a leaf term, -r(t), and then plus,

405 00:16:54.690 --> 00:16:56.940 here, this is essentially a coupling,

406 00:16:56.940 --> 00:16:59.963 all of this is inside here, the brackets with a plus,

407 00:16:59.963 --> 00:17:01.920 this is like a (indistinct) function,

408 00:17:01.920 --> 00:17:03.840 so this is just taking the positive part

409 00:17:03.840 --> 00:17:05.160 of what's on the inside.

410 00:17:05.160 --> 00:17:07.590 Here, b is an activation threshold,

411 00:17:07.590 --> 00:17:10.860 W is a wiring matrix, and then r are those rates again.

412 00:17:10.860 --> 00:17:13.200 And then C here, that's essentially covariants

413 00:17:13.200 --> 00:17:16.590 for some noise term perturbing this process.

414 00:17:16.590 --> 00:17:19.260 We use these TLNs to test the sensitivity

415 00:17:19.260 --> 00:17:20.820 of our transfer entropy estimators

416 00:17:20.820 --> 00:17:23.730 to common and private noise sources as you change C,

417 00:17:23.730 --> 00:17:27.180 as well as how well the transfer entropy network agrees

418 00:17:27.180 --> 00:17:29.433 with the wiring matrix.

419 00:17:30.720 --> 00:17:33.300 A particular class of TLNs that were really nice

420 00:17:33.300 --> 00:17:35.010 for these experiments are called

421 00:17:35.010 --> 00:17:36.990 combinatorial threshold linear networks.

422 00:17:36.990 --> 00:17:38.070 These are really pretty new,

423 00:17:38.070 --> 00:17:42.270 these were introduced by Carina Curto's lab this year.

424 00:17:42.270 --> 00:17:46.500 And really, this was inspired by a talk I'd seen her give

425 00:17:46.500 --> 00:17:49.110 at FACM in May.

426 00:17:49.110 --> 00:17:50.820 These are threshold linear networks

427 00:17:50.820 --> 00:17:52.320 where the weight matrix here, W,

428 00:17:52.320 --> 00:17:55.440 representing the wiring of the neurons,

429 00:17:55.440 --> 00:17:58.020 is determined by a directed graph G.

430 00:17:58.020 --> 00:17:59.610 So you start with some directed graph G,

431 00:17:59.610 --> 00:18:00.810 that's what's shown here on the left.

432 00:18:00.810 --> 00:18:02.910 This figure is adapted from Carina's paper,

433 00:18:02.910 --> 00:18:03.743 this is a very nice paper

434 00:18:03.743 --> 00:18:05.470 if you'd like to take a look at it.

435 00:18:06.690 --> 00:18:09.003 And if i and j are not connected,

436 00:18:10.020 --> 00:18:12.030 then the weight matrix is assigned one value,

437 00:18:12.030 --> 00:18:14.460 and if they are connected, then it's assigned another value.

438 00:18:14.460 --> 00:18:18.300 And the wiring is zero if i equals j.

439 00:18:18.300 --> 00:18:20.430 These networks are nice if we want to test

440 00:18:20.430 --> 00:18:23.820 structural hypotheses, because it's very easy to predict

441 00:18:23.820 --> 00:18:26.820 from the input graph how the output dynamics

442 00:18:26.820 --> 00:18:28.170 of the network should behave.

443 00:18:28.170 --> 00:18:29.610 They're a really beautiful analysis

444 00:18:29.610 --> 00:18:31.530 that Carina does in this paper to show

445 00:18:31.530 --> 00:18:32.940 that you can produce all these different

446 00:18:32.940 --> 00:18:34.890 interlocking patterns of limit cycles,

447 00:18:34.890 --> 00:18:36.990 and multi-step states, and chaos,

448 00:18:36.990 --> 00:18:38.220 and all these nice patterns,

449 00:18:38.220 --> 00:18:39.330 and you can design them

450 00:18:39.330 --> 00:18:42.723 by picking these nice directed graphs.

451 00:18:43.890 --> 00:18:46.230 The last class of networks that we've built to test

452 00:18:46.230 --> 00:18:47.760 are leaky-integrate and fire networks.

453 00:18:47.760 --> 00:18:51.000 So here, we're using a leaky integrate and fire model,

454 00:18:51.000 --> 00:18:54.390 where our wiring matrix W is drawn randomly,

455 00:18:54.390 --> 00:18:56.580 it's block-stochastic,

456 00:18:56.580 --> 00:18:59.820 which means that it's Erdos-Renyi between blocks.

457 00:18:59.820 --> 00:19:02.010 And it's a balanced network,

458 00:19:02.010 --> 00:19:04.200 so we have excitatory and inhibitory neurons

459 00:19:04.200 --> 00:19:08.100 that talk to each other and maintain a balance

460 00:19:08.100 --> 00:19:09.210 in the dynamics here.

461 00:19:09.210 --> 00:19:11.340 The hope is to pick a large enough scale network

462 00:19:11.340 --> 00:19:13.380 that we see properly chaotic dynamics

463 00:19:13.380 --> 00:19:15.480 using this leaky integrate and fire model.

464 00:19:17.340 --> 00:19:20.760 These tests have yielded fairly mixed results.

465 00:19:20.760 --> 00:19:23.610 So the simple tests behave as expected.

466 00:19:23.610 --> 00:19:26.760 So the estimators that are used are biased,

467 00:19:26.760 --> 00:19:28.560 and the bias typically decays slower

468 00:19:28.560 --> 00:19:30.030 than the variance estimation,

469 00:19:30.030 --> 00:19:32.490 which means that you do need fairly long trajectories

470 00:19:32.490 --> 00:19:36.240 to try to properly estimate the transfer entropy.

471 00:19:36.240 --> 00:19:38.430 That said, transfer entropy does correctly identify

472 00:19:38.430 --> 00:19:40.320 causal relationships in simple graphs,

473 00:19:40.320 --> 00:19:43.980 and transfer entropy matches the underlying structure

474 00:19:43.980 --> 00:19:47.550 used in combinatorial threshold linear networks, so CTLN.

475 00:19:48.810 --> 00:19:52.200 Unfortunately, these results did not carry over as cleanly

476 00:19:52.200 --> 00:19:54.180 to the leaky integrate and fire models,

477 00:19:54.180 --> 00:19:56.070 or to larger models.

478 00:19:56.070 --> 00:19:58.410 So what I'm showing you on the right here,

479 00:19:58.410 --> 00:20:00.240 this is a matrix where we've calculated

480 00:20:00.240 --> 00:20:01.500 the pairwise transfer entropy

481 00:20:01.500 --> 00:20:06.240 between all neurons in a 150 neuron balanced network.

482 00:20:06.240 --> 00:20:09.390 This is shown absolute, this is shown in the log scale.

483 00:20:09.390 --> 00:20:11.280 And the main thing I want to highlight, first,

484 00:20:11.280 --> 00:20:12.390 taking a look at this matrix,

485 00:20:12.390 --> 00:20:15.030 it's very hard to see exactly what the structure is.

486 00:20:15.030 --> 00:20:16.530 You see this banding?

487 00:20:16.530 --> 00:20:19.830 That's because neurons tend to be highly predictive

488 00:20:19.830 --> 00:20:20.790 if they fire a lot.

489 00:20:20.790 --> 00:20:22.020 So there's a strong correlation

490 00:20:22.020 --> 00:20:25.410 between the transfer entropy between x and y,

491 00:20:25.410 --> 00:20:27.603 and just the activity level of x.

492 00:20:28.860 --> 00:20:31.170 But it's hard to distinguish blockwise differences,

493 00:20:31.170 --> 00:20:34.290 for example, between inhibitory neurons, excitatory neurons,

494 00:20:34.290 --> 00:20:35.760 and that really takes plotting out,

495 00:20:35.760 --> 00:20:38.640 so here, this box and whisker plot on the bottom,

496 00:20:38.640 --> 00:20:42.540 this is showing you if we group entries of this matrix

497 00:20:42.540 --> 00:20:43.530 by type of connection.

498 00:20:43.530 --> 00:20:45.371 So maybe excitatory to excitatory,

499 00:20:45.371 --> 00:20:48.120 or inhibitor to excitatory, or so on,

500 00:20:48.120 --> 00:20:50.160 that the distribution of realized transfer entropy

501 00:20:50.160 --> 00:20:52.050 is really different,

502 00:20:52.050 --> 00:20:54.120 but they're different in sort of subtle ways.

503 00:20:54.120 --> 00:20:57.273 So in this larger scale balanced network,

504 00:20:58.110 --> 00:21:02.370 it's much less clear whether transfer entropy

505 00:21:02.370 --> 00:21:05.160 effectively is equated in some way

506 00:21:05.160 --> 00:21:07.803 with the true connectivity or wiring.

507 00:21:08.760 --> 00:21:10.230 In some ways, this is not a surprise,

508 00:21:10.230 --> 00:21:11.760 because the behavior of the balanced networks

509 00:21:11.760 --> 00:21:12.840 is inherently balanced,

510 00:21:12.840 --> 00:21:15.750 and Erdos-Renyi is inherently in the structure.

511 00:21:15.750 --> 00:21:19.110 But there are ways in which these experiments have revealed

512 00:21:19.110 --> 00:21:22.290 confounding factors that are conceptual factors

513 00:21:22.290 --> 00:21:25.410 that make transfer entropies not an ideal measure,

514 00:21:25.410 --> 00:21:27.510 or maybe not as ideal as it seems,

515 00:21:27.510 --> 00:21:29.400 given the start of this talk.

516 00:21:29.400 --> 00:21:33.450 So for example, suppose two trajectories X and Y

517 00:21:33.450 --> 00:21:36.090 are both strongly driven by a third trajectory Z,

518 00:21:36.090 --> 00:21:38.520 but X responds to Z first.

519 00:21:38.520 --> 00:21:40.380 Well, then the present information about X,

520 00:21:40.380 --> 00:21:41.460 or the present state of X,

521 00:21:41.460 --> 00:21:43.230 carries information about the future of Y,

522 00:21:43.230 --> 00:21:45.000 so X is predictive of Y.

523 00:21:45.000 --> 00:21:47.280 So X forecasts Y, so in the transfer entropy

524 00:21:47.280 --> 00:21:50.790 or Wiener causality setting, we would say X causes Y,

525 00:21:50.790 --> 00:21:53.133 even if X and Y are only both responding to Z.

526 00:21:54.480 --> 00:21:57.750 So here, in this example, suppose you have a directed tree

527 00:21:57.750 --> 00:22:02.100 where information or dynamics propagate down the tree.

528 00:22:02.100 --> 00:22:06.570 If you look at this node here, Pj and i,

529 00:22:06.570 --> 00:22:10.920 Pj will react to essentially information

530 00:22:10.920 --> 00:22:13.230 traveling down this tree before i does,

531 00:22:13.230 --> 00:22:15.270 so Pj would be predictive for i,

532 00:22:15.270 --> 00:22:18.510 so we would observe an effective connection,

533 00:22:18.510 --> 00:22:20.670 where Pj forecasts i.

534 00:22:20.670 --> 00:22:22.650 Which means that neurons that are not directly connected

535 00:22:22.650 --> 00:22:25.920 may influence each other, and that this transfer entropy,

536 00:22:25.920 --> 00:22:28.500 really, you should think of in terms of forecasting,

537 00:22:28.500 --> 00:22:32.103 not in terms of being a direct analog to the wiring matrix.

538 00:22:33.270 --> 00:22:34.980 One way around this is to condition

539 00:22:34.980 --> 00:22:36.870 on the state of the rest of the network

540 00:22:36.870 --> 00:22:38.520 before you start doing some averaging.

541 00:22:38.520 --> 00:22:40.890 This leads to some other notions of entropy,

542 00:22:40.890 --> 00:22:42.450 so, for example, causation entropy,

543 00:22:42.450 --> 00:22:43.800 and this is sort of a promising direction,

544 00:22:43.800 --> 00:22:47.310 but it's not a direction we've had time to explore yet.

545 00:22:47.310 --> 00:22:49.260 So that's the estimation side.

546 00:22:49.260 --> 00:22:51.630 Those are the tools for estimating transfer entropy.

547 00:22:51.630 --> 00:22:52.800 Now, let's switch gears

548 00:22:52.800 --> 00:22:55.170 and talk about that second question I introduced,

549 00:22:55.170 --> 00:22:57.450 which is essentially, how do we analyze structure.

550 00:22:57.450 --> 00:23:00.450 Suppose we could calculate a transfer entropy graph.

551 00:23:00.450 --> 00:23:03.600 How would we extract structural information from that graph?

552 00:23:03.600 --> 00:23:06.240 And here, I'm going to be introducing some tools

553 00:23:06.240 --> 00:23:07.530 that I've worked on for a while

554 00:23:07.530 --> 00:23:11.370 for describing random structures and graphs.

555 00:23:11.370 --> 00:23:14.700 These are tied back to some work I've really done

556 00:23:14.700 --> 00:23:17.730 as a graduate student, and conversations with Lek-Heng.

557 00:23:17.730 --> 00:23:19.320 So we start in a really simple context,

558 00:23:19.320 --> 00:23:20.670 we just have a graph or network.

559 00:23:20.670 --> 00:23:22.560 This could be directed or undirected,

560 00:23:22.560 --> 00:23:23.790 and we're gonna require that it does not have self-loops,

561 00:23:23.790 --> 00:23:25.650 and that it's finite.

562 00:23:25.650 --> 00:23:27.930 We'll let V here be the number of vertices,

563 00:23:27.930 --> 00:23:30.390 and E be the number of edges.

564 00:23:30.390 --> 00:23:32.730 Then the object of study that we'll introduce

565 00:23:32.730 --> 00:23:34.020 is something called an edge flow.

566 00:23:34.020 --> 00:23:35.340 An edge flow is essentially a function

567 00:23:35.340 --> 00:23:36.810 on the edges of the graph.

568 00:23:36.810 --> 00:23:39.870 So this is a function that accepts pairs of endpoints

569 00:23:39.870 --> 00:23:41.580 and returns a real number.

570 00:23:41.580 --> 00:23:42.990 And this is an alternating function,

571 00:23:42.990 --> 00:23:46.710 so if I take f(i, j), that's negative f(j, i),

572 00:23:46.710 --> 00:23:49.350 because you can think of f(i, j) as being some flow,

573 00:23:49.350 --> 00:23:51.810 like a flow of material between i and j,

574 00:23:51.810 --> 00:23:53.910 hence this name, edge flow.

575 00:23:53.910 --> 00:23:55.620 This is analogous to a vector field,

576 00:23:55.620 --> 00:23:58.140 because this is analogous to the structure of a vector field

577 00:23:58.140 --> 00:23:58.973 on the graph,

578 00:23:58.973 --> 00:24:02.084 and represents some sort of flow between nodes.

579 00:24:02.084 --> 00:24:04.440 Edge flows are really sort of generic things.

580 00:24:04.440 --> 00:24:06.900 So you can take this idea of an edge flow

581 00:24:06.900 --> 00:24:08.910 and apply it in a lot of different areas,

582 00:24:08.910 --> 00:24:09.870 because really all you need

583 00:24:09.870 --> 00:24:11.970 is you just need the structure of some alternating function

584 00:24:11.970 --> 00:24:13.410 on the edges of a graph.

585 00:24:13.410 --> 00:24:16.140 So I've read papers,

586 00:24:16.140 --> 00:24:18.570 and worked in a bunch of these different areas.

587 00:24:18.570 --> 00:24:20.640 Particularly, I've focused on applications of this

588 00:24:20.640 --> 00:24:24.660 in game theory, in pairwise and social choice settings,

589 00:24:24.660 --> 00:24:26.130 in biology and Markov chains.

590 00:24:26.130 --> 00:24:28.170 And a lot of this project has been attempting

591 00:24:28.170 --> 00:24:31.320 to take this experience working with edge flows in,

592 00:24:31.320 --> 00:24:34.140 for example, say, non-equilibrium thermodynamics,

593 00:24:34.140 --> 00:24:35.940 or looking at pairwise preference data,

594 00:24:35.940 --> 00:24:37.830 and looking at a different application area

595 00:24:37.830 --> 00:24:39.630 here to neuroscience.

596 00:24:39.630 --> 00:24:41.580 Really, you can you think about the edge flow,

597 00:24:41.580 --> 00:24:43.170 or relevant edge flow in neuroscience,

598 00:24:43.170 --> 00:24:45.780 you might be asking about asymmetries in wiring patterns,

599 00:24:45.780 --> 00:24:48.840 or differences in directed influence or causality,

600 00:24:48.840 --> 00:24:50.070 or, really, you can think about

601 00:24:50.070 --> 00:24:51.270 these transfer entropy quantities.

602 00:24:51.270 --> 00:24:53.010 This is why I was excited about transfer entropy.

603 00:24:53.010 --> 00:24:55.770 Transfer entropy is inherently directed notion

604 00:24:55.770 --> 00:24:59.070 of information flow, so it's natural to think that

605 00:24:59.070 --> 00:25:01.380 if you can calculate things like the transfer entropy,

606 00:25:01.380 --> 00:25:03.540 then really, what you're studying is some sort of edge flow

607 00:25:03.540 --> 00:25:04.373 on a graph.

608 00:25:05.820 --> 00:25:10.200 Edge flows often are subject to the same common questions.

609 00:25:10.200 --> 00:25:12.150 So if I want to analyze the structure of an edge flow,

610 00:25:12.150 --> 00:25:13.770 there's some really big global questions

611 00:25:13.770 --> 00:25:15.120 that I would often ask,

612 00:25:15.120 --> 00:25:17.920 that get asked in all these different application areas.

613 00:25:19.140 --> 00:25:20.340 One common question is,

614 00:25:20.340 --> 00:25:22.710 well, does the flow originate somewhere and end somewhere?

615 00:25:22.710 --> 00:25:25.020 Are there sources and sinks in the graph?

616 00:25:25.020 --> 00:25:26.067 Another is, does it circulate?

617 00:25:26.067 --> 00:25:29.073 And if it does circulate, on what scales, and where?

618 00:25:30.720 --> 00:25:32.520 If you have a network that's connected

619 00:25:32.520 --> 00:25:34.890 to a whole exterior network, for example,

620 00:25:34.890 --> 00:25:36.540 if you're looking at some small subsystem

621 00:25:36.540 --> 00:25:38.310 that's embedded in a much larger system,

622 00:25:38.310 --> 00:25:40.710 as is almost always the case in neuroscience,

623 00:25:40.710 --> 00:25:42.000 then you also need to think about

624 00:25:42.000 --> 00:25:43.290 what passes through the network.

625 00:25:43.290 --> 00:25:44.970 So, is there a flow or current

626 00:25:44.970 --> 00:25:46.647 that moves through the boundary of the network,

627 00:25:46.647 --> 00:25:50.070 and is there information that flows through

628 00:25:50.070 --> 00:25:52.230 the network that you're studying?

629 00:25:52.230 --> 00:25:54.660 And in particular, if we have these different types of flow,

630 00:25:54.660 --> 00:25:56.640 if flow can originate in source and end in sinks,

631 00:25:56.640 --> 00:25:59.040 if it can circulate, if it can pass through,

632 00:25:59.040 --> 00:26:02.550 can we decompose the flow into pieces that do each of these

633 00:26:02.550 --> 00:26:04.983 and ask how much of the flow does 1, 2, or 3?

634 00:26:06.810 --> 00:26:09.333 Those questions lead to a decomposition.

635 00:26:10.590 --> 00:26:13.470 So here, we're going to start with a simple idea.

636 00:26:13.470 --> 00:26:14.940 We're going to decompose an edge flow

637 00:26:14.940 --> 00:26:17.430 by projecting it onto orthogonal subspaces

638 00:26:17.430 --> 00:26:20.040 associated with some graph operators.

639 00:26:20.040 --> 00:26:23.023 Generically, if we consider two linear operators,

640 00:26:23.023 --> 00:26:26.760 A and B, where the product A times B equals zero,

641 00:26:26.760 --> 00:26:29.160 then the range of B must be contained

642 00:26:29.160 --> 00:26:31.350 in the null space of A,

643 00:26:31.350 --> 00:26:33.420 which means that I can express

644 00:26:33.420 --> 00:26:34.950 essentially any set of real numbers,

645 00:26:34.950 --> 00:26:36.330 so you can think of this as being

646 00:26:36.330 --> 00:26:39.360 the vector space of possible edge flows,

647 00:26:39.360 --> 00:26:42.690 as a direct sum of the range of B,

648 00:26:42.690 --> 00:26:44.730 the range of A transpose,

649 00:26:44.730 --> 00:26:47.007 and the intersection of the null space of B transpose

650 00:26:47.007 --> 00:26:48.420 and the null space of A.

651 00:26:48.420 --> 00:26:52.680 This blue subspace, this is called the harmonic space,

652 00:26:52.680 --> 00:26:57.680 and this is trivial in many applications

653 00:26:57.810 --> 00:26:59.790 if you choose A and B correctly.

654 00:26:59.790 --> 00:27:02.220 So there's often settings where you can pick A and B

655 00:27:02.220 --> 00:27:05.700 so that these two null spaces have no intersection,

656 00:27:05.700 --> 00:27:07.860 and then this decomposition boils down

657 00:27:07.860 --> 00:27:12.860 to just separating a vector space into the range of B

658 00:27:12.900 --> 00:27:14.373 and the range of A transpose.

659 00:27:15.780 --> 00:27:17.820 In the graph setting, our goal is essentially

660 00:27:17.820 --> 00:27:20.430 to pick these operators to be meaningful things,

661 00:27:20.430 --> 00:27:21.900 that is, to pick graph operators

662 00:27:21.900 --> 00:27:25.890 so that these subspaces carry a meaningful,

663 00:27:25.890 --> 00:27:29.700 or carry meaning in the structural context.

664 00:27:29.700 --> 00:27:33.480 So let's think a little bit about graph operators here.

665 00:27:33.480 --> 00:27:35.490 So, let's look at two different classes of operators.

666 00:27:35.490 --> 00:27:40.350 So we can consider matrices that have E rows and n columns,

667 00:27:40.350 --> 00:27:43.410 or matrices that have l rows and E columns,

668 00:27:43.410 --> 00:27:46.010 where again, E is the number of edges in this graph.

669 00:27:47.790 --> 00:27:50.190 If I have a matrix with E rows,

670 00:27:50.190 --> 00:27:53.370 then each column with a matrix has as many entries

671 00:27:53.370 --> 00:27:54.960 as there are edges in the graph,

672 00:27:54.960 --> 00:27:57.420 so it can be thought of as itself an edge flow.

673 00:27:57.420 --> 00:27:58.530 So you can think that this matrix

674 00:27:58.530 --> 00:28:00.120 is composed of a set of columns,

675 00:28:00.120 --> 00:28:03.150 where each column is some particular motivic flow,

676 00:28:03.150 --> 00:28:04.173 or flow motif.

677 00:28:05.430 --> 00:28:09.450 In contrast, if I look at a matrix where I have E columns,

678 00:28:09.450 --> 00:28:11.430 then each row of the matrix is a flow motif,

679 00:28:11.430 --> 00:28:15.900 so products against M evaluate inner products

680 00:28:15.900 --> 00:28:18.360 against specific flow motifs.

681 00:28:18.360 --> 00:28:19.620 That means in this context,

682 00:28:19.620 --> 00:28:21.090 if I look at the range of this matrix,

683 00:28:21.090 --> 00:28:22.710 this is really a linear combination

684 00:28:22.710 --> 00:28:25.230 of a specific subset of flow motifs,

685 00:28:25.230 --> 00:28:26.340 and in this context,

686 00:28:26.340 --> 00:28:27.780 if I look at the null space of the matrix,

687 00:28:27.780 --> 00:28:30.030 I'm looking at all edge flows orthogonal

688 00:28:30.030 --> 00:28:32.040 to that set of flow motifs.

689 00:28:32.040 --> 00:28:36.240 So here, if I look at the range of a matrix with E rows,

690 00:28:36.240 --> 00:28:38.730 that subspace is essentially modeling behavior

691 00:28:38.730 --> 00:28:41.670 similar to the motifs, so if I pick a set of motifs

692 00:28:41.670 --> 00:28:45.180 that flow out of a node, or flow into a node,

693 00:28:45.180 --> 00:28:48.180 then this range is going to be a subspace of edge flows

694 00:28:48.180 --> 00:28:51.330 that tend to originate in sources and end in sinks.

695 00:28:51.330 --> 00:28:53.790 In contrast, here, the null space of M,

696 00:28:53.790 --> 00:28:56.910 that's all edge flows orthogonal to the flow motifs,

697 00:28:56.910 --> 00:28:59.010 so it models behavior distinct from the motifs.

698 00:28:59.010 --> 00:29:02.490 Essentially, this space asks what doesn't the flow do,

699 00:29:02.490 --> 00:29:04.803 whereas this space asks what does the flow do.

700 00:29:06.540 --> 00:29:09.180 Here is a simple, very classical example.

701 00:29:09.180 --> 00:29:11.040 And this goes all the way back to, you can think,

702 00:29:11.040 --> 00:29:13.710 like Kirchhoff electric circuit theory.

703 00:29:13.710 --> 00:29:15.180 We can define two operators.

704 00:29:15.180 --> 00:29:17.850 Here, G, this is essentially a gradient operator.

705 00:29:17.850 --> 00:29:20.430 And if you've taken some graph theory, you might know this

706 00:29:20.430 --> 00:29:22.320 as the edge incidence matrix.

707 00:29:22.320 --> 00:29:24.930 This is the matrix which essentially records

708 00:29:24.930 --> 00:29:26.400 the endpoints of an edge,

709 00:29:26.400 --> 00:29:29.100 and evaluates differences across it.

710 00:29:29.100 --> 00:29:32.760 So for example, if I look at this first row of G,

711 00:29:32.760 --> 00:29:35.340 this corresponds to edge I in the graph,

712 00:29:35.340 --> 00:29:38.670 and if I had a function defined on the nodes in the graph,

713 00:29:38.670 --> 00:29:42.780 products with G would evaluate differences across this edge.

714 00:29:42.780 --> 00:29:44.340 If you look at its columns,

715 00:29:44.340 --> 00:29:45.930 each column here is a flow motif.

716 00:29:45.930 --> 00:29:48.900 So for example, this highlighted second column,

717 00:29:48.900 --> 00:29:51.510 this is entries 1, -1, 0, -1,

718 00:29:51.510 --> 00:29:53.070 if you carry those back to the edges,

719 00:29:53.070 --> 00:29:56.100 that corresponds to this specific flow motif.

720 00:29:56.100 --> 00:29:58.400 So here, this gradient, it's adjoint,

721 00:29:58.400 --> 00:30:00.300 so essentially a divergence operator,

722 00:30:00.300 --> 00:30:03.300 which means that the flow motifs are unit in flows

723 00:30:03.300 --> 00:30:05.190 or unit out flows for specific nodes,

724 00:30:05.190 --> 00:30:07.170 like what's shown here.

725 00:30:07.170 --> 00:30:09.540 You can also introduce something like a curl operator.

726 00:30:09.540 --> 00:30:13.200 The curl operator evaluates path sums around loops.

727 00:30:13.200 --> 00:30:16.170 So this row here, for example, this is a flow motif

728 00:30:16.170 --> 00:30:20.430 corresponding to the loop labeled A in this graph.

729 00:30:20.430 --> 00:30:22.050 You could certainly imagine other operators

730 00:30:22.050 --> 00:30:23.400 build other motifs.

731 00:30:23.400 --> 00:30:25.020 These operators are particularly nice,

732 00:30:25.020 --> 00:30:27.070 because they define principled subspaces.

733 00:30:28.200 --> 00:30:30.990 So if we apply that generic decomposition,

734 00:30:30.990 --> 00:30:34.140 then we could say that the space of possible edge flows, RE,

735 00:30:34.140 --> 00:30:37.410 can be decomposed into the range of the gradient operator,

736 00:30:37.410 --> 00:30:39.480 the range of the curl transpose,

737 00:30:39.480 --> 00:30:41.640 and the intersection of their null spaces,

738 00:30:41.640 --> 00:30:43.770 into this harmonic space.

739 00:30:43.770 --> 00:30:45.810 This is nice, because the range of the gradient,

740 00:30:45.810 --> 00:30:47.730 that's flows that start and end somewhere,

741 00:30:47.730 --> 00:30:49.350 those are flows that are associated

742 00:30:49.350 --> 00:30:51.990 with motion (indistinct) potential.

743 00:30:51.990 --> 00:30:53.220 So these, if you're thinking physics,

744 00:30:53.220 --> 00:30:54.630 you might say that these are conservative,

745 00:30:54.630 --> 00:30:56.520 these are flows generated by voltage

746 00:30:56.520 --> 00:30:58.680 if you're looking at an electric circuit.

747 00:30:58.680 --> 00:31:01.410 These cyclic flows, while these are the flows and range

748 00:31:01.410 --> 00:31:03.840 of the curl transpose, and then this harmonic space,

749 00:31:03.840 --> 00:31:06.360 those are flows that enter and leave the network

750 00:31:06.360 --> 00:31:09.960 without either starting or ending at a sink or a source,

751 00:31:09.960 --> 00:31:11.040 or circulating.

752 00:31:11.040 --> 00:31:11.940 So you can think that really,

753 00:31:11.940 --> 00:31:14.460 this decomposes the space of edge flows

754 00:31:14.460 --> 00:31:17.220 into flows that start and end somewhere inside the network,

755 00:31:17.220 --> 00:31:19.110 flows that circulate within the network,

756 00:31:19.110 --> 00:31:20.310 and flows that do neither,

757 00:31:20.310 --> 00:31:22.470 i.e. flows that enter and leave the network.

758 00:31:22.470 --> 00:31:25.140 So this accomplishes that initial decomposition

759 00:31:25.140 --> 00:31:26.390 I'd set out at the start.

760 00:31:28.110 --> 00:31:29.400 Once we have this decomposition,

761 00:31:29.400 --> 00:31:32.580 then we can evaluate the sizes of the components

762 00:31:32.580 --> 00:31:36.300 of the decomposition to measure how much of the flow

763 00:31:36.300 --> 00:31:39.300 starts and ends somewhere, how much circulates, and so on.

764 00:31:39.300 --> 00:31:41.370 So, we can introduce these generic measures,

765 00:31:41.370 --> 00:31:43.023 where given some operator M,

766 00:31:44.100 --> 00:31:45.960 we decompose the space of edge flows

767 00:31:45.960 --> 00:31:49.020 into the range of M and the null space of M transpose,

768 00:31:49.020 --> 00:31:52.050 which means we can project f onto these subspaces,

769 00:31:52.050 --> 00:31:54.570 and then evaluate the sizes of these components,

770 00:31:54.570 --> 00:31:57.570 and that's a way of measuring how much of the flow

771 00:31:57.570 --> 00:32:00.630 behaves like the flow motifs contained in this operator,

772 00:32:00.630 --> 00:32:01.680 and how much doesn't.

773 00:32:04.080 --> 00:32:04.920 So, yeah.

774 00:32:04.920 --> 00:32:06.690 So that lets us answer this question,

775 00:32:06.690 --> 00:32:09.150 and this is the tool that we're going to be using

776 00:32:09.150 --> 00:32:10.893 as our measurable.

777 00:32:12.270 --> 00:32:15.510 Now, that's totally easy to do,

778 00:32:15.510 --> 00:32:17.240 if you're given a fixed edge flow and a fixed graph.

779 00:32:17.240 --> 00:32:19.380 If you have a fixed graph, you can build your operators,

780 00:32:19.380 --> 00:32:21.630 you choose the motifs, you have fixed edge flow,

781 00:32:21.630 --> 00:32:24.030 you just project the edge flow onto the subspaces,

782 00:32:24.030 --> 00:32:26.910 span by those operators, and you're done.

783 00:32:26.910 --> 00:32:29.730 However, there are many cases where

784 00:32:29.730 --> 00:32:32.850 it's worth thinking about a distribution of edge flows,

785 00:32:32.850 --> 00:32:35.913 and then expected structures given that distribution.

786 00:32:36.780 --> 00:32:39.120 So here, we're going to be considering random edge flows,

787 00:32:39.120 --> 00:32:40.740 for example, an edge flow of capital F.

788 00:32:40.740 --> 00:32:43.350 Here, I'm using capital letters to denote random quantities

789 00:32:43.350 --> 00:32:44.850 sampled from an edge flow distribution.

790 00:32:44.850 --> 00:32:47.268 So this is the distribution of possible edge flows.

791 00:32:47.268 --> 00:32:48.360 And this is worth thinking about

792 00:32:48.360 --> 00:32:51.480 because many generative models are stochastic.

793 00:32:51.480 --> 00:32:52.980 They may involve some random seed,

794 00:32:52.980 --> 00:32:54.870 or they may, for example, like that neural model,

795 00:32:54.870 --> 00:32:57.780 or a lot of these sort of neural models, be chaotic,

796 00:32:57.780 --> 00:33:01.050 so even if they are deterministic generative models,

797 00:33:01.050 --> 00:33:03.270 the output data behaves as though it's been sampled

798 00:33:03.270 --> 00:33:04.270 from a distribution.

799 00:33:05.430 --> 00:33:07.020 On the empirical side, for example,

800 00:33:07.020 --> 00:33:09.030 when we're estimating transfer entropy,

801 00:33:09.030 --> 00:33:11.070 or estimating some information flow,

802 00:33:11.070 --> 00:33:13.380 then there's always some degree of measurement error,

803 00:33:13.380 --> 00:33:15.420 or uncertainty in the estimate,

804 00:33:15.420 --> 00:33:17.520 which really means that from a Bayesian perspective,

805 00:33:17.520 --> 00:33:19.720 we should be thinking that our estimator

806 00:33:20.580 --> 00:33:23.580 is a point estimate drawn from some posterior distribution

807 00:33:23.580 --> 00:33:25.260 of edge flows, and that we're back in the setting where,

808 00:33:25.260 --> 00:33:27.780 again, we need to talk about a distribution.

809 00:33:27.780 --> 00:33:30.720 Lastly, this random edge flow setting is also

810 00:33:30.720 --> 00:33:33.723 really important if we want to compare the null hypotheses.

811 00:33:34.740 --> 00:33:36.990 Because often, if you want to compare

812 00:33:36.990 --> 00:33:38.370 to some sort of null hypothesis,

813 00:33:38.370 --> 00:33:40.920 it's helpful to have an ensemble of edge flows

814 00:33:40.920 --> 00:33:42.540 to compare against,

815 00:33:42.540 --> 00:33:44.370 which means that we would like to be able to talk about

816 00:33:44.370 --> 00:33:47.763 expected structure under varying distributional assumptions.

817 00:33:49.650 --> 00:33:54.210 If we can talk meaningfully about random edge flows,

818 00:33:54.210 --> 00:33:56.100 then really what we can start doing

819 00:33:56.100 --> 00:33:58.920 is we can start bridging the expected structure

820 00:33:58.920 --> 00:34:00.240 back to the distribution.

821 00:34:00.240 --> 00:34:01.290 So what we're looking for

822 00:34:01.290 --> 00:34:04.620 is a way of explaining generic expectations

823 00:34:04.620 --> 00:34:06.990 of what the structure will look like

824 00:34:06.990 --> 00:34:09.690 as we vary this distribution of edge flows.

825 00:34:09.690 --> 00:34:12.720 You could think that a particular dynamical system

826 00:34:12.720 --> 00:34:17.720 generates a wiring pattern, that generates firing dynamics,

827 00:34:19.260 --> 00:34:20.730 those firing dynamics determine

828 00:34:20.730 --> 00:34:23.190 some sort of information flow graph,

829 00:34:23.190 --> 00:34:24.690 and then that information flow graph

830 00:34:24.690 --> 00:34:27.750 is really a sample from that generative model,

831 00:34:27.750 --> 00:34:30.480 and we would like to be able to talk about

832 00:34:30.480 --> 00:34:31.680 what would we expect

833 00:34:31.680 --> 00:34:33.840 if we knew the distribution of edge flows

834 00:34:33.840 --> 00:34:35.310 about the global structure.

835 00:34:35.310 --> 00:34:36.960 That is, we'd like to bridge global structure

836 00:34:36.960 --> 00:34:38.670 back to this distribution.

837 00:34:38.670 --> 00:34:40.950 And then, ideally, you'd bridge that distribution

838 00:34:40.950 --> 00:34:42.390 back to the generative mechanism.

839 00:34:42.390 --> 00:34:44.670 And this is a project for future work.

840 00:34:44.670 --> 00:34:46.650 Obviously, this is fairly ambitious.

841 00:34:46.650 --> 00:34:49.150 However, this first point is something you can do

842 00:34:50.610 --> 00:34:53.040 really in fairly explicit detail,

843 00:34:53.040 --> 00:34:54.180 and that's what I would like to spell out

844 00:34:54.180 --> 00:34:55.290 with the end of this talk,

845 00:34:55.290 --> 00:34:58.080 is how do you bridge global structure

846 00:34:58.080 --> 00:34:59.943 back to a distribution of edge flows.

847 00:35:02.220 --> 00:35:03.480 So yeah.

848 00:35:03.480 --> 00:35:04.500 So that's our main question.

849 00:35:04.500 --> 00:35:06.210 How does the choice of distribution

850 00:35:06.210 --> 00:35:08.553 influence the expected global flow structure?

851 00:35:12.000 --> 00:35:14.790 So first, let's start with a lemma.

852 00:35:14.790 --> 00:35:17.010 Suppose that we have a distribution of edge flows

853 00:35:17.010 --> 00:35:19.920 with some expectation f bar, and some covariance,

854 00:35:19.920 --> 00:35:23.640 here I'm using double bar V to denote covariance.

855 00:35:23.640 --> 00:35:26.710 We'll let S contained in the set of...

856 00:35:26.710 --> 00:35:29.340 S will be a subspace contained within

857 00:35:29.340 --> 00:35:31.110 the vector space of edge flows,

858 00:35:31.110 --> 00:35:35.100 and we'll let PS be the orthogonal projector onto S.

859 00:35:35.100 --> 00:35:40.100 Then FS, that's the projection of F onto this subspace S,

860 00:35:40.140 --> 00:35:42.900 the expectation of its norm squared

861 00:35:42.900 --> 00:35:47.900 is the norm of the expected flow projected onto S squared,

862 00:35:48.390 --> 00:35:51.760 so this is essentially the expectation of the sample

863 00:35:52.680 --> 00:35:55.800 is the measure evaluated with the expected sample.

864 00:35:55.800 --> 00:35:58.140 And then plus a term that involves an inner product

865 00:35:58.140 --> 00:36:00.240 between the projector on the subspace

866 00:36:00.240 --> 00:36:02.160 and the covariance matrix for the edge flows.

867 00:36:02.160 --> 00:36:03.960 Here, this denotes the matrix inner product,

868 00:36:03.960 --> 00:36:06.993 so is just the sum over all ij entries.

869 00:36:09.030 --> 00:36:10.470 What's nice about this formula is,

870 00:36:10.470 --> 00:36:12.780 at least in terms of expectation,

871 00:36:12.780 --> 00:36:17.010 it reduces this study of the bridge

872 00:36:17.010 --> 00:36:19.890 between distribution and network structure

873 00:36:19.890 --> 00:36:21.660 to a study of moments, right?

874 00:36:21.660 --> 00:36:23.520 Because we've replaced a distributional problem here

875 00:36:23.520 --> 00:36:26.730 with a linear algebra problem

876 00:36:26.730 --> 00:36:28.740 that's posed in terms of this projector,

877 00:36:28.740 --> 00:36:30.570 the projector out of the subspace S,

878 00:36:30.570 --> 00:36:33.360 which is determined by the topology of the network.

879 00:36:33.360 --> 00:36:35.760 And the variance in that edge flow,

880 00:36:35.760 --> 00:36:38.010 which is determined by your generative model.

881 00:36:39.660 --> 00:36:42.150 Well, you might say, "Okay, well, fine,

882 00:36:42.150 --> 00:36:43.920 this is a matrix inner product, we can just stop here.

883 00:36:43.920 --> 00:36:45.000 We could compute this projector.

884 00:36:45.000 --> 00:36:47.027 We could sample a whole bunch of edge flows

885 00:36:47.027 --> 00:36:48.068 to compute this covariance.

886 00:36:48.068 --> 00:36:50.040 So you can do this matrix inner product."

887 00:36:50.040 --> 00:36:53.580 But I'm sort of greedy, because I suspect

888 00:36:53.580 --> 00:36:57.480 that you can really do more with this inner product.

889 00:36:57.480 --> 00:36:59.500 So I'd like to highlight some challenges

890 00:37:00.360 --> 00:37:02.760 associated with this inner product.

891 00:37:02.760 --> 00:37:05.670 So first, let's say I asked you to design a distribution

892 00:37:05.670 --> 00:37:07.350 with tuneable global structure.

893 00:37:07.350 --> 00:37:09.063 So for example, I said I want you to

894 00:37:09.063 --> 00:37:10.170 pick a generative model,

895 00:37:10.170 --> 00:37:12.060 or design a distribution of edge flows,

896 00:37:12.060 --> 00:37:14.040 that when I sample edge flows from it,

897 00:37:14.040 --> 00:37:18.360 their expected structures match some expectation.

898 00:37:18.360 --> 00:37:20.910 It's not obvious how to do that given this formula.

899 00:37:21.750 --> 00:37:24.150 It's not obvious in particular, because these projectors,

900 00:37:24.150 --> 00:37:26.160 like the projector onto subspace S,

901 00:37:26.160 --> 00:37:28.590 typically depend in fairly non-trivial ways

902 00:37:28.590 --> 00:37:29.910 on the graph topology.

903 00:37:29.910 --> 00:37:31.650 So small changes in the graph topology

904 00:37:31.650 --> 00:37:34.350 can completely change this projector.

905 00:37:34.350 --> 00:37:37.350 In essence, it's hard to isolate topology from distribution.

906 00:37:37.350 --> 00:37:38.790 You could think that this inner product,

907 00:37:38.790 --> 00:37:41.313 if I think about it in terms of the ij entries,

908 00:37:43.110 --> 00:37:46.560 while easy to compute, is not easy to interpret,

909 00:37:46.560 --> 00:37:49.470 because i and j are somewhat arbitrary indexing.

910 00:37:49.470 --> 00:37:51.330 And obviously, really, the topology of the graph,

911 00:37:51.330 --> 00:37:53.160 it's not encoded in the indexing,

912 00:37:53.160 --> 00:37:56.160 it's encoded in the structure of these matrices.

913 00:37:56.160 --> 00:37:57.420 So in some ways, what we really need

914 00:37:57.420 --> 00:38:00.003 is a better basis for computing this inner product.

915 00:38:01.320 --> 00:38:03.090 In addition, computing this inner product

916 00:38:03.090 --> 00:38:05.280 just may not be empirically feasible,

917 00:38:05.280 --> 00:38:06.510 because it might not be feasible

918 00:38:06.510 --> 00:38:07.860 to estimate all these covariances.

919 00:38:07.860 --> 00:38:09.240 There's lots of settings where,

920 00:38:09.240 --> 00:38:10.740 if you have a random edge flow,

921 00:38:10.740 --> 00:38:12.900 it becomes very expensive to try to estimate

922 00:38:12.900 --> 00:38:14.850 all the covariances in this graph, or sorry,

923 00:38:14.850 --> 00:38:18.570 in this matrix, because this matrix has as many entries

924 00:38:18.570 --> 00:38:20.793 as there are pairs of edges in the graph.

925 00:38:22.110 --> 00:38:25.650 And typically, that number of edges grows fairly quickly

926 00:38:25.650 --> 00:38:27.300 in the number of nodes in the graph.

927 00:38:27.300 --> 00:38:30.630 So in the worst case, the size of these matrices

928 00:38:30.630 --> 00:38:33.330 goes not to the square of the number of nodes in the graph,

929 00:38:33.330 --> 00:38:34.950 but the number of nodes in the graph to the fourth,

930 00:38:34.950 --> 00:38:37.380 so this becomes very expensive very fast.

931 00:38:37.380 --> 00:38:40.590 Again, we could try to address this problem

932 00:38:40.590 --> 00:38:43.410 if we had a better basis for performing this inner product,

933 00:38:43.410 --> 00:38:45.780 because we might hope to be able to truncate

934 00:38:45.780 --> 00:38:47.040 somewhere in that basis,

935 00:38:47.040 --> 00:38:49.190 and use a lower dimensional representation.

936 00:38:50.160 --> 00:38:51.630 So, to build there,

937 00:38:51.630 --> 00:38:54.930 I'm going to show you a particular family of covariances.

938 00:38:54.930 --> 00:38:58.230 We're going to start with a very simple generative model.

939 00:38:58.230 --> 00:39:00.300 So let's suppose that each node of the graph

940 00:39:00.300 --> 00:39:01.860 is assigned some set of attributes,

941 00:39:01.860 --> 00:39:04.382 here, a random vector X, sampled from a...

942 00:39:04.382 --> 00:39:05.250 So you can think of trait space,

943 00:39:05.250 --> 00:39:07.080 a space of possible attributes.

944 00:39:07.080 --> 00:39:08.970 And these are sampled i.i.d.

945 00:39:08.970 --> 00:39:10.980 In addition, we'll assume there exists

946 00:39:10.980 --> 00:39:12.930 an alternating function f,

947 00:39:12.930 --> 00:39:15.360 which accepts pairs of attributes,

948 00:39:15.360 --> 00:39:17.130 and returns a real number.

949 00:39:17.130 --> 00:39:20.070 So this is something that I can evaluate on the endpoints

950 00:39:20.070 --> 00:39:22.683 of an edge, and return an edge flow value.

951 00:39:24.420 --> 00:39:26.340 In this setting,

952 00:39:26.340 --> 00:39:29.160 everything that I've shown you before simplifies.

953 00:39:29.160 --> 00:39:31.740 So if my edge flow F is drawn

954 00:39:31.740 --> 00:39:33.780 by first sampling a set of attributes,

955 00:39:33.780 --> 00:39:36.090 and then plugging those attributes into functions

956 00:39:36.090 --> 00:39:41.090 on the edges, then the mean edge flow is zero,

957 00:39:41.880 --> 00:39:43.800 so that f bar goes away,

958 00:39:43.800 --> 00:39:46.080 and the covariance reduces to this form.

959 00:39:46.080 --> 00:39:47.100 So you get a standard form,

960 00:39:47.100 --> 00:39:49.260 where the covariance and the edge flow

961 00:39:49.260 --> 00:39:51.840 is a function of two scalar quantities,

962 00:39:51.840 --> 00:39:53.010 that's sigma squared and rho,

963 00:39:53.010 --> 00:39:56.400 these are both statistics associated with this function

964 00:39:56.400 --> 00:39:59.220 and the distribution of traits.

965 00:39:59.220 --> 00:40:01.560 And then some matrices, so we have an identity matrix,

966 00:40:01.560 --> 00:40:04.620 and we have this gradient matrix showing up again.

967 00:40:04.620 --> 00:40:07.160 This is really nice, because when you plug it back in,

968 00:40:07.160 --> 00:40:08.400 we try to compute, say,

969 00:40:08.400 --> 00:40:11.403 the expected sizes of the components,

970 00:40:12.510 --> 00:40:14.880 this matrix inner product

971 00:40:14.880 --> 00:40:16.920 that I was complaining about before,

972 00:40:16.920 --> 00:40:19.290 this whole matrix inner product simplifies.

973 00:40:19.290 --> 00:40:21.060 So when you have a variance

974 00:40:21.060 --> 00:40:23.400 that's in this nice, simple, canonical form,

975 00:40:23.400 --> 00:40:25.800 then the expected overall size of the edge flow,

976 00:40:25.800 --> 00:40:28.620 that's just sigma squared, the expected size

977 00:40:28.620 --> 00:40:31.353 projected onto that conservative subspace,

978 00:40:32.250 --> 00:40:34.830 that breaks into this combination

979 00:40:34.830 --> 00:40:36.840 of the sigma squared and the rho,

980 00:40:36.840 --> 00:40:38.940 again, those are some simple statistics.

981 00:40:38.940 --> 00:40:42.360 And then V, E, L, and E, those are just

982 00:40:42.360 --> 00:40:44.040 essentially dimension counting on the network.

983 00:40:44.040 --> 00:40:46.860 So this is the number of vertices, the number of edges,

984 00:40:46.860 --> 00:40:48.480 and the number of loops, the number of loops,

985 00:40:48.480 --> 00:40:49.320 that's the number of edges

986 00:40:49.320 --> 00:40:51.990 minus the number of vertices plus one.

987 00:40:51.990 --> 00:40:54.720 And similarly, the expected cyclic size,

988 00:40:54.720 --> 00:40:57.240 or size of the cyclic component, reduces to,

989 00:40:57.240 --> 00:41:00.660 again, this scalar factor in terms of the statistics,

990 00:41:00.660 --> 00:41:05.643 and some dimension counting topology related quantities.

991 00:41:07.762 --> 00:41:08.790 So this is very nice,

992 00:41:08.790 --> 00:41:11.610 because this allows us to really separate

993 00:41:11.610 --> 00:41:14.280 the role of topology from the role of the generative model.

994 00:41:14.280 --> 00:41:16.980 The generative model determines sigma and rho,

995 00:41:16.980 --> 00:41:19.323 and topology determines these dimensions.

996 00:41:21.630 --> 00:41:24.280 It turns out that the same thing is true

997 00:41:25.560 --> 00:41:28.590 even if you don't sample the edge flow

998 00:41:28.590 --> 00:41:32.610 using this trait approach, but the graph is complete.

999 00:41:32.610 --> 00:41:34.380 So if your graph is complete,

1000 00:41:34.380 --> 00:41:36.630 then no matter how you sample your edge flow,

1001 00:41:36.630 --> 00:41:38.280 for any edge flow distribution,

1002 00:41:38.280 --> 00:41:40.350 exactly the same formulas hold,

1003 00:41:40.350 --> 00:41:42.840 you just replace those simple statistics

1004 00:41:42.840 --> 00:41:44.760 with estimators for those statistics

1005 00:41:44.760 --> 00:41:46.770 given your sampled flow.

1006 00:41:46.770 --> 00:41:48.900 And this is sort of a striking result,

1007 00:41:48.900 --> 00:41:51.150 because this says that this conclusion

1008 00:41:51.150 --> 00:41:53.730 that was linked to some specific generative model

1009 00:41:53.730 --> 00:41:55.740 with some very specific assumptions, right,

1010 00:41:55.740 --> 00:41:57.330 we assumed it was i.i.d.,

1011 00:41:57.330 --> 00:41:59.100 extends to all complete graphs,

1012 00:41:59.100 --> 00:42:02.193 regardless of the actual distribution that we sampled from.

1013 00:42:04.650 --> 00:42:05.790 Up until this point,

1014 00:42:05.790 --> 00:42:07.790 this is kind of just an algebra miracle.

1015 00:42:09.180 --> 00:42:10.950 And one of the things I'd like to do at the end of this talk

1016 00:42:10.950 --> 00:42:12.660 is explain why this is true,

1017 00:42:12.660 --> 00:42:14.823 and show how to generalize these results.

1018 00:42:16.080 --> 00:42:16.950 So to build there,

1019 00:42:16.950 --> 00:42:19.050 let's emphasize some of the advantages of this.

1020 00:42:19.050 --> 00:42:21.540 So first, the advantages of the model,

1021 00:42:21.540 --> 00:42:23.970 it's mechanistically plausible in certain settings, 1022 00:42:23.970 --> 00:42:27.510 it cleanly separated the role of topology and distribution,

1023 00:42:27.510 --> 00:42:29.880 and these coefficients that had to do with topology,

1024 00:42:29.880 --> 00:42:30.960 these are just dimensions,

1025 00:42:30.960 --> 00:42:33.510 these are non negative quantities,

1026 00:42:33.510 --> 00:42:36.030 so it's easy to work out monotonic relationships

1027 00:42:36.030 --> 00:42:39.690 between expected structure and simple statistics

1028 00:42:39.690 --> 00:42:41.190 of the edge flow distribution.

1029 00:42:43.770 --> 00:42:47.010 The fact that you can do that enables more general analysis.

1030 00:42:47.010 --> 00:42:48.240 So I'm showing you on the right here,

1031 00:42:48.240 --> 00:42:50.730 this is from a different application area.

1032 00:42:50.730 --> 00:42:55.140 This was an experiment where we trained a set of agents

1033 00:42:55.140 --> 00:42:57.600 to play a game using a genetic algorithm,

1034 00:42:57.600 --> 00:42:59.970 and then we looked at the expected sizes

1035 00:42:59.970 --> 00:43:02.400 of cyclic and acyclic components

1036 00:43:02.400 --> 00:43:04.770 in a tournament among those agents.

1037 00:43:04.770 --> 00:43:07.620 And you can actually predict these curves

1038 00:43:07.620 --> 00:43:09.780 using this type of structure analysis,

1039 00:43:09.780 --> 00:43:13.230 because it was possible to predict the dynamics

1040 00:43:13.230 --> 00:43:16.713 of these simple statistics, this sigma and this rho.

1041 00:43:17.730 --> 00:43:19.980 So this is a really powerful analytical tool,

1042 00:43:19.980 --> 00:43:22.530 but it is limited to this particular model.

1043 00:43:22.530 --> 00:43:25.590 In particular, it only models unstructured cycles,

1044 00:43:25.590 --> 00:43:26.970 so if you look at the cyclic component

1045 00:43:26.970 --> 00:43:28.350 generated by this model, 1046 00:43:28.350 --> 00:43:30.990 it just looks like random noise that's been projected

1047 00:43:30.990 --> 00:43:32.990 onto the range of the current transpose.

1048 00:43:33.870 --> 00:43:36.120 It's limited to correlations on adjacent edges,

1049 00:43:36.120 --> 00:43:37.890 so we only generate correlations

1050 00:43:37.890 --> 00:43:39.960 on edges that share an endpoint, because you could think

1051 00:43:39.960 --> 00:43:41.850 that all of the original random information

1052 00:43:41.850 --> 00:43:43.233 comes from the endpoints.

1053 00:43:44.490 --> 00:43:46.560 And then, in some ways, it's not general enough.

1054 00:43:46.560 --> 00:43:48.060 So it lacks an expressivity.

1055 00:43:48.060 --> 00:43:50.970 We can't parameterize all possible expected structures

1056 00:43:50.970 --> 00:43:54.270 by picking sigma and rho.

1057 00:43:54.270 --> 00:43:55.920 And we lack some notion of sufficiency,

1058 00:43:55.920 --> 00:43:58.410 i.e. if the graph is not complete,

1059 00:43:58.410 --> 00:44:00.840 then this nice algebraic property,

1060 00:44:00.840 --> 00:44:02.970 that it actually didn't matter what the distribution was,

1061 00:44:02.970 --> 00:44:04.470 this fails to hold.

1062 00:44:04.470 --> 00:44:06.060 So if the graph is not complete,

1063 00:44:06.060 --> 00:44:09.312 then projection onto the family of covariances

1064 00:44:09.312 --> 00:44:11.430 parameterized in this fashion

1065 00:44:11.430 --> 00:44:13.473 changes the expected global structure.

1066 00:44:14.640 --> 00:44:16.980 So we would like to address these limitations.

1067 00:44:16.980 --> 00:44:18.810 And so our goal for the next part of this talk

1068 00:44:18.810 --> 00:44:21.240 is to really generalize these results.

1069 00:44:21.240 --> 00:44:22.230 To generalize,

1070 00:44:22.230 --> 00:44:24.930 we're going to switch our perspective a little bit.

1071 00:44:24.930 --> 00:44:27.420 So I'll recall this formula,

1072 00:44:27.420 --> 00:44:29.730 that if we generate our edge flow

1073 00:44:29.730 --> 00:44:31.650 by sampling quantities on the endpoints,

1074 00:44:31.650 --> 00:44:34.110 and then plugging them into functions on the edges,

1075 00:44:34.110 --> 00:44:35.297 then you necessarily get a covariance

1076 00:44:35.297 --> 00:44:37.320 that's in this two parameter family,

1077 00:44:37.320 --> 00:44:38.820 where I have two scalar quantities

1078 00:44:38.820 --> 00:44:40.590 associated with the statistics of the edge flow,

1079 00:44:40.590 --> 00:44:42.210 that's this sigma and this rho,

1080 00:44:42.210 --> 00:44:43.440 and then I have some matrices

1081 00:44:43.440 --> 00:44:45.480 that are associated with the topology of the network

1082 00:44:45.480 --> 00:44:47.463 in the subspaces I'm projecting onto.

1083 00:44:48.480 --> 00:44:50.760 These are related to a different way

1084 00:44:50.760 --> 00:44:52.290 of looking at the graph.

1085 00:44:52.290 --> 00:44:54.450 So I can start with my original graph,

1086 00:44:54.450 --> 00:44:56.760 and then I can convert it to an edge graph,

1087 00:44:56.760 --> 00:44:59.373 where I have one node per edge in the graph,

1088 00:45:00.210 --> 00:45:02.823 and nodes are connected if they share an endpoint.

1089 00:45:04.080 --> 00:45:07.320 You can then assign essentially signs to these edges

1090 00:45:07.320 --> 00:45:10.110 based on whether the edge direction

1091 00:45:10.110 --> 00:45:13.710 chosen in the original graph is consistent or inconsistent

1092 00:45:13.710 --> 00:45:15.810 at the node that links two edges.

1093 00:45:15.810 --> 00:45:19.890 So for example, edges 1 and 2 both point in to this node,

1094 00:45:19.890 --> 00:45:21.780 so there's an edge linking 1 and 2

1095 00:45:21.780 --> 00:45:24.540 in the edge graph with a positive sign.

1096 00:45:24.540 --> 00:45:25.470 This essentially tells you

1097 00:45:25.470 --> 00:45:30.150 that the influence of random information

1098 00:45:30.150 --> 00:45:33.240 assigned on this node linking 1 and 2

1099 00:45:33.240 --> 00:45:36.210 would positively correlate the sample edge flow

1100 00:45:36.210 --> 00:45:37.323 on edges 1 and 2.

1101 00:45:38.370 --> 00:45:42.990 Then, this form, what this form for covariance matrices says

1102 00:45:42.990 --> 00:45:46.200 is that we're looking at families of edge flows

1103 00:45:46.200 --> 00:45:48.690 that have correlations on edges sharing an endpoint,

1104 00:45:48.690 --> 00:45:51.150 so edges at distance one in this edge graph,

1105 00:45:51.150 --> 00:45:52.290 and non-adjacent edges

1106 00:45:52.290 --> 00:45:54.240 are entirely independent of each other.

1107 00:45:56.310 --> 00:45:57.143 Okay.

1108 00:45:58.230 --> 00:46:00.330 So that's essentially what the trait-performance model

1109 00:46:00.330 --> 00:46:01.693 is doing, is it's parameterizing

1110 00:46:01.693 --> 00:46:03.690 a family of covariance matrices,

1111 00:46:03.690 --> 00:46:05.910 where we're modeling correlations at distance one,

1112 00:46:05.910 --> 00:46:07.590 but not further in the edge graph.

1113 00:46:07.590 --> 00:46:08.820 So then the natural thought

1114 00:46:08.820 --> 00:46:10.717 for how to generalize these results is to ask,

1115 00:46:10.717 --> 00:46:13.677 "Can we model longer distance correlations to this graph?"

1116 00:46:15.000 --> 00:46:16.590 To do so, let's think a little bit

1117 00:46:16.590 --> 00:46:19.260 about what this matrix

1118 00:46:19.260 --> 00:46:20.970 that's showing up inside the covariances,

1119 00:46:20.970 --> 00:46:23.820 so we have a gradient times a gradient transpose.

1120 00:46:23.820 --> 00:46:27.903 This is in effect a Laplacian for that edge graph.

1121 00:46:29.700 --> 00:46:31.680 And you can do this for other motifs.

1122 00:46:31.680 --> 00:46:34.710 If you think about different motif constructions, 1123 00:46:34.710 --> 00:46:38.400 essentially if you take a product of M transpose times M,

1124 00:46:38.400 --> 00:46:41.070 that will generate something that looks like a Laplacian

1125 00:46:41.070 --> 00:46:44.070 or an adjacency matrix for a graph

1126 00:46:44.070 --> 00:46:47.250 where I'm assigning nodes to be motifs,

1127 00:46:47.250 --> 00:46:50.190 and looking at the overlap of motifs.

1128 00:46:50.190 --> 00:46:51.990 And if I look at M times M transpose,

1129 00:46:51.990 --> 00:46:54.840 and I'm looking at the overlap of edges via shared motifs.

1130 00:46:54.840 --> 00:46:57.300 So these operators you can think about as being Laplacians

1131 00:46:57.300 --> 00:46:58.650 for some sort of graph

1132 00:46:58.650 --> 00:47:01.413 that's generated from the original graph motifs.

1133 00:47:03.630 --> 00:47:06.450 Like any adjacency matrix,

1134 00:47:06.450 --> 00:47:11.040 powers of something like G, G transpose minus 2I,

1135 00:47:11.040 --> 00:47:13.800 that would model connections along longer paths,

1136 00:47:13.800 --> 00:47:15.810 along longer distances in these graphs

1137 00:47:15.810 --> 00:47:18.710 associated with motifs, in this case, with the edge graph.

1138 00:47:19.620 --> 00:47:21.240 So our thought is, maybe, well,

1139 00:47:21.240 --> 00:47:22.890 we could extend this trait performance

1140 00:47:22.890 --> 00:47:24.630 family of covariance matrices

1141 00:47:24.630 --> 00:47:26.610 by instead of only looking at

1142 00:47:26.610 --> 00:47:30.750 a linear combination of an identity matrix and this matrix,

1143 00:47:30.750 --> 00:47:32.190 we could look at a power series.

1144 00:47:32.190 --> 00:47:36.600 So we could consider combining powers of this matrix.

1145 00:47:36.600 --> 00:47:39.390 And this would generate this family of matrices

1146 00:47:39.390 --> 00:47:40.800 that are parameterized by some set of

1147 00:47:40.800 --> 00:47:43.080 coefficients (indistinct)... <v ->Dr. Strang.</v>

1148 00:47:43.080 --> 00:47:45.600 I apologize, I just wanted to remind you

1149 00:47:45.600 --> 00:47:48.240 that we have a rather tight time limit,

1150 00:47:48.240 --> 00:47:50.250 approximately a couple of minutes.

1151 00:47:50.250 --> 00:47:51.303 <v ->Yes, of course.</v>

1152 00:47:52.170 --> 00:47:57.150 So here, the idea is to parameterize this family of matrices

1153 00:47:57.150 --> 00:48:00.450 by introducing a set of polynomials with coefficients alpha,

1154 00:48:00.450 --> 00:48:03.420 and then plugging into the polynomial

1155 00:48:03.420 --> 00:48:06.000 the Laplacian that's generated by...

1156 00:48:06.000 --> 00:48:09.000 The adjacency matrix generated by the graph motifs

1157 00:48:09.000 --> 00:48:10.830 we're interested in.

1158 00:48:10.830 --> 00:48:12.030 And that trait performance result,

1159 00:48:12.030 --> 00:48:14.310 that was really just looking at the first order case here,

1160 00:48:14.310 --> 00:48:17.070 that was looking at a linear polynomial

1161 00:48:17.070 --> 00:48:19.680 with these chosen coefficients.

1162 00:48:19.680 --> 00:48:24.120 This power series model is really nice analytically.

1163 00:48:24.120 --> 00:48:28.260 So if we start with some graph operator M,

1164 00:48:28.260 --> 00:48:31.020 and we consider the family of covariance matrices

1165 00:48:31.020 --> 00:48:34.260 generated by plugging M, M transpose into some

1166 00:48:34.260 --> 00:48:36.240 polynomial and power series,

1167 00:48:36.240 --> 00:48:38.520 then this family of matrices

1168 00:48:38.520 --> 00:48:42.213 is contained within the span of powers of M, M transpose.

1169 00:48:45.030 --> 00:48:47.970 You can talk about this family in terms of combinatorics.

1170 00:48:47.970 --> 00:48:49.830 So, for example, if we use that gradient

1171 00:48:49.830 --> 00:48:52.410 times gradient transpose minus twice the identity,

1172 00:48:52.410 --> 00:48:54.660 then powers of this is essentially, again, path counting,

1173 00:48:54.660 --> 00:48:56.673 so this is counting paths of length n.

1174 00:48:57.780 --> 00:49:00.270 You can also look at things like the trace of these powers.

1175 00:49:00.270 --> 00:49:01.980 So if you look at the trace series,

1176 00:49:01.980 --> 00:49:05.310 that's the sequence where you look at the trace of powers

1177 00:49:05.310 --> 00:49:07.893 of these, essentially, these adjacency matrices.

1178 00:49:08.820 --> 00:49:10.770 This is doing some sort of loop count,

1179 00:49:10.770 --> 00:49:13.800 where we're counting loops of different length.

1180 00:49:13.800 --> 00:49:15.300 And you can think of this trace series, in some sense,

1181 00:49:15.300 --> 00:49:18.690 as controlling amplification of self-correlations

1182 00:49:18.690 --> 00:49:20.140 within the sampled edge flow.

1183 00:49:21.840 --> 00:49:22.980 Depending on the generative model,

1184 00:49:22.980 --> 00:49:24.720 we might want to use different operators

1185 00:49:24.720 --> 00:49:26.070 for generating these families.

1186 00:49:26.070 --> 00:49:29.160 So for example, going back to that synaptic plasticity model

1187 00:49:29.160 --> 00:49:32.820 with coupled oscillators, in this case, using the gradient

1188 00:49:32.820 --> 00:49:35.010 to generate the family of covariance matrices

1189 00:49:35.010 --> 00:49:36.750 is not really the right structure,

1190 00:49:36.750 --> 00:49:39.690 because the dynamics of the model

1191 00:49:39.690 --> 00:49:42.690 have these natural cyclic connections.

1192 00:49:42.690 --> 00:49:45.660 So it's better to build the power series using the curl.

1193 00:49:45.660 --> 00:49:47.130 So depending on your model,

1194 00:49:47.130 --> 00:49:48.840 you can adapt this power series family

1195 00:49:48.840 --> 00:49:50.940 by plugging in a different graph operator.

1196 00:49:52.560 --> 00:49:55.200 Let's see now what happens if we try to compute

1197 00:49:55.200 --> 00:49:57.810 the expected sizes of some components

1198 00:49:57.810 --> 00:50:00.240 using a power series of this form.

1199 00:50:00.240 --> 00:50:04.380 So, if the variance, or covariance matrix for edge flow

1200 00:50:04.380 --> 00:50:06.270 is a power series in, for example,

1201 00:50:06.270 --> 00:50:08.460 the gradient, gradient transpose,

1202 00:50:08.460 --> 00:50:11.580 then the expected sizes of the measures

1203 00:50:11.580 --> 00:50:14.460 can all be expressed as linear combinations

1204 00:50:14.460 --> 00:50:16.110 of this trace series

1205 00:50:16.110 --> 00:50:18.600 and the coefficients of the original polynomial.

1206 00:50:18.600 --> 00:50:21.390 For example, the expected cyclic size of the flow

1207 00:50:21.390 --> 00:50:23.700 is just the polynomial evaluated at negative two,

1208 00:50:23.700 --> 00:50:26.130 multiplied by the number of loops in the graph.

1209 00:50:26.130 --> 00:50:27.840 And this, this really generalizes

1210 00:50:27.840 --> 00:50:29.040 that trait performance result,

1211 00:50:29.040 --> 00:50:30.150 because the trait performance result

1212 00:50:30.150 --> 00:50:33.033 is given by restricting these polynomials to be linear.

1213 00:50:36.270 --> 00:50:39.693 This, you can extend to other bases.

1214 00:50:41.310 --> 00:50:43.260 But really, what this accomplishes

1215 00:50:43.260 --> 00:50:45.210 is by generalizing trait performance,

1216 00:50:45.210 --> 00:50:50.210 we achieve this generic properties that it failed to have.

1217 00:50:52.140 --> 00:50:55.560 So in particular, if I have an edge flow subspace S

1218 00:50:55.560 --> 00:50:58.740 spanned by the flow motifs stored in some operator M,

1219 00:50:58.740 --> 00:51:00.840 then this power series family of covariances

1220 00:51:00.840 --> 00:51:05.190 associated with the Laplacian, that is, M times M transpose,

1221 00:51:05.190 --> 00:51:08.160 is both expressive, in the sense that

1222 00:51:08.160 --> 00:51:10.950 for any non negative a and b,

1223 00:51:10.950 --> 00:51:13.380 I can pick some alpha and beta

1224 00:51:13.380 --> 00:51:14.730 so that the expected size

1225 00:51:14.730 --> 00:51:17.700 of the projection of F onto the subspaces a,

1226 00:51:17.700 --> 00:51:21.600 and the projected size of F on the subspace orthogonal to S

1227 00:51:21.600 --> 00:51:26.133 is b for any covariance in this power series family.

1228 00:51:27.060 --> 00:51:29.760 And it's sufficient in the sense that

1229 00:51:29.760 --> 00:51:32.160 for any edge flow distribution with mean zero,

1230 00:51:32.160 --> 00:51:34.710 and covariance V,

1231 00:51:34.710 --> 00:51:37.980 if C is the matrix nearest to V in Frobenius norm,

1232 00:51:37.980 --> 00:51:40.380 restricted to the power series family,

1233 00:51:40.380 --> 00:51:43.770 then these inner products computed in terms of C

1234 00:51:43.770 --> 00:51:45.570 are exactly the same as inner products

1235 00:51:45.570 --> 00:51:47.070 computed in terms of V,

1236 00:51:47.070 --> 00:51:49.020 so they directly predict the structure,

1237 00:51:49.020 --> 00:51:51.390 which means that if I use this power series family,

1238 00:51:51.390 --> 00:51:53.580 discrepancies off of this family

1239 00:51:53.580 --> 00:51:55.380 don't change the expected structure.

1240 00:51:56.520 --> 00:51:57.353 Okay.

1241 00:51:57.353 --> 00:51:59.010 So, I know I'm short on time here,

1242 00:51:59.010 --> 00:52:02.790 so I'd like to skip, then, just to the end of this talk.

1243 00:52:02.790 --> 00:52:04.200 There's further things you can do with this,

1244 00:52:04.200 --> 00:52:06.660 this is sort of really nice mathematically.

1245 00:52:06.660 --> 00:52:09.510 You can build an approximation theory out of this,

1246 00:52:09.510 --> 00:52:11.730 and study it for different random graph families,

1247 00:52:11.730 --> 00:52:14.820 how many terms in these power series you need.

1248 00:52:14.820 --> 00:52:16.380 And those terms define

1249 00:52:16.380 --> 00:52:18.570 some nicer simple minimal set of statistics,

1250 00:52:18.570 --> 00:52:20.433 to try to estimate structure.

1251 00:52:22.110 --> 00:52:25.350 But I'd like to really just get to the end here,

1252 00:52:25.350 --> 00:52:28.260 and emphasize the takeaways from this talk.

1253 00:52:28.260 --> 00:52:29.580 So the first half of this talk

1254 00:52:29.580 --> 00:52:32.130 was focused on information flow.

1255 00:52:32.130 --> 00:52:35.160 What we saw is that information flow is a non-trivial,

1256 00:52:35.160 --> 00:52:36.810 but well studied estimation problem.

1257 00:52:36.810 --> 00:52:38.280 And this is something that, at least on my side,

1258 00:52:38.280 --> 00:52:40.530 is a work in progress with students.

1259 00:52:40.530 --> 00:52:42.150 Here, the, in some ways,

1260 00:52:42.150 --> 00:52:43.380 the conclusion of that first half

1261 00:52:43.380 --> 00:52:44.820 would be that causation entropy

1262 00:52:44.820 --> 00:52:46.890 may be a more appropriate measure than TE

1263 00:52:46.890 --> 00:52:48.540 when trying to build these flow graphs

1264 00:52:48.540 --> 00:52:51.240 to apply these structural measures to.

1265 00:52:51.240 --> 00:52:53.730 Then, on the structural side, we can say that

1266 00:52:53.730 --> 00:52:54.600 power series families,

1267 00:52:54.600 --> 00:52:56.610 this is a nice family of covariance matrices.

1268 00:52:56.610 --> 00:52:59.490 It has nice properties that are useful empirically, 1269 00:52:59.490 --> 00:53:01.830 because they let us build global correlation structures

1270 00:53:01.830 --> 00:53:03.450 from a sequence of local correlations

1271 00:53:03.450 --> 00:53:04.683 from that power series.

1272 00:53:06.240 --> 00:53:08.220 If you plug this back into the expected measures,

1273 00:53:08.220 --> 00:53:09.990 you can recover monotonic relations,

1274 00:53:09.990 --> 00:53:12.180 like in that limited trait performance case.

1275 00:53:12.180 --> 00:53:14.400 And truncation of these power series

1276 00:53:14.400 --> 00:53:15.870 reduces the number of quantities

1277 00:53:15.870 --> 00:53:17.663 that you would actually need to measure.

1278 00:53:18.600 --> 00:53:21.210 Actually, to a number of quantities that can be quite small

1279 00:53:21.210 --> 00:53:22.080 relative to the graph,

1280 00:53:22.080 --> 00:53:24.353 and that's where this approximation theory comes in.

1281 00:53:25.260 --> 00:53:28.140 One way, maybe to sort of summarize this entire approach,

1282 00:53:28.140 --> 00:53:30.810 is what we've done is by looking at these power series

1283 00:53:30.810 --> 00:53:33.030 built in terms of the graph operators,

1284 00:53:33.030 --> 00:53:35.460 is it provides a way to study

1285 00:53:35.460 --> 00:53:39.120 inherently heterogeneous connections, or covariances,

1286 00:53:39.120 --> 00:53:40.530 or edge flows distributions,

1287 00:53:40.530 --> 00:53:42.630 using a homogeneous correlation model

1288 00:53:42.630 --> 00:53:46.110 that's built at multiple scales by starting with local scale

1289 00:53:46.110 --> 00:53:47.553 and then looking at powers.

1290 00:53:48.960 --> 00:53:50.340 In some ways, this is a common...

1291 00:53:50.340 --> 00:53:53.310 I ended a previous version of this talk with,

1292 00:53:53.310 --> 00:53:55.110 I still think that this structural analysis is,

1293 00:53:55.110 --> 00:53:57.270 in some ways, a hammer seeking a nail, 1294 00:53:57.270 --> 00:53:59.160 and that this information flow construction,

1295 00:53:59.160 --> 00:54:02.100 this is work in progress to try and build that nail.

1296 00:54:02.100 --> 00:54:04.110 So thank you all for your attention.

1297 00:54:04.110 --> 00:54:06.690 I'll turn it now over to questions.

1298 00:54:06.690 --> 00:54:08.784 <v Instructor>(indistinct)</v>

1299 00:54:08.784 --> 00:54:11.370 Thank you so much for your talk.

1300 00:54:11.370 --> 00:54:12.573 Really appreciate it.

1301 00:54:14.610 --> 00:54:15.600 For those of you on Zoom,

1302 00:54:15.600 --> 00:54:17.400 you're welcome to keep up the conversations,

1303 00:54:17.400 --> 00:54:19.890 but unfortunately we have to clear the room,

1304 00:54:19.890 --> 00:54:21.330 so I do apologize.

1305 00:54:21.330 --> 00:54:22.230 But, (indistinct).

1306 00:54:24.690 --> 00:54:25.523 Dr. Strang?

1307 00:54:26.480 --> 00:54:27.423 Am I muted?

1308 00:54:30.330 --> 00:54:31.560 Dr. Strang?

1309 00:54:31.560 --> 00:54:33.190 <v ->Oh, yes, yeah.</v>

1310 00:54:33.190 --> 00:54:35.160 <v Instructor>Okay, do you mind if people...</v>

1311 00:54:35.160 --> 00:54:36.960 We have to clear the room, do you mind if people

1312 00:54:36.960 --> 00:54:38.610 email you if they have questions?

1313 00:54:39.990 --> 00:54:42.060 <v ->I'm sorry, I couldn't hear the end of the question.</v>

1314 00:54:42.060 --> 00:54:43.130 Do I mind if...

1315 00:54:45.060 --> 00:54:46.530 <v Instructor>We have to clear the room,</v>

1316 00:54:46.530 --> 00:54:48.990 do you mind if people email you if they have questions,

1317 00:54:48.990 --> 00:54:51.037 and (indistinct)... <v ->No, no, not at all.</v>

1318 00:54:51.933 --> 00:54:54.466 <v Instructor>So I do apologize, they are literally</v>

1319 00:54:54.466 --> 00:54:56.760 (indistinct) the room right now.

1320 00:54:56.760 --> 00:54:59.100 <v ->Okay, no, yeah, that's totally fine.</v>

1321 00:54:59.100 --> 00:55:00.660 <v Instructor>Thank you.</v>

1322 00:55:00.660 --> 00:55:02.820 And thanks again for a wonderful talk.

1323 00:55:02.820 --> 00:55:03.653 <v ->Thank you.</v>