WEBVTT

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

2 00:00:00.990 --> 00:00:04.131 In respect for everybody's time today,

3 00:00:04.131 --> 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 and 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 --> 00:00:25.413 Born in Ohio, so representer.

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 --> 00:00:41.580 where he is a William H. Kruskal Instructor of Statistics,

16 00:00:41.580 --> 00:00:43.470 and Applied Mathematics.

17 00:00:43.470 --> 00:00:46.680 Today he is 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 --> 00:00:57.780 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 a 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 --> 00:01:22.290 This project more generally was born

32 00:01:22.290 --> 00:01:24.450 out of conversations with Brent Doiron

33 00:01:24.450 --> 00:01:27.330 and Lek-Heng Lim here at Chicago.

34 00:01:27.330 --> 00:01:29.700 Brent really was the inspiration for

35 00:01:29.700 --> 00:01:32.610 starting to venture into computational neuroscience.

36 00:01:32.610 --> 00:01:35.430 I'll really say that that I am new to this world,

37 00:01:35.430 --> 00:01:36.750 it's a world that's exciting to me,

38 00:01:36.750 --> 00:01:40.920 but really is a world that I am actively exploring

39 00:01:40.920 --> 00:01:41.753 and learning about.

40 00:01:41.753 --> 00:01:44.400 So I look forward to conversations afterwards

41 00:01:44.400 --> 00:01:46.170 to learn more here.

42 00:01:46.170 --> 00:01:49.440 My background was much more inspired by Lek-Heng's work

43 00:01:49.440 --> 00:01:50.973 in computational topology.

44 00:01:52.380 --> 00:01:54.300 And some of what I'll be presenting today

45 00:01:54.300 --> 00:01:56.553 is really inspired by conversations with him.

46 00:01:57.690 --> 00:02:00.340 So let's start with some introduction and motivation.

47 00:02:01.200 --> 00:02:03.273 The motivation personally for this talk,

48 00:02:04.620 --> 00:02:06.420 so it goes back really to work that I started

49 00:02:06.420 --> 00:02:07.800 when I was a graduate student,

50 00:02:07.800 --> 00:02:09.810 I've had this sort of long standing interest

51 00:02:09.810 --> 00:02:12.300 in the interplay between structure and dynamics,

52 00:02:12.300 --> 00:02:14.430 in particular on networks.

53 00:02:14.430 --> 00:02:15.570 I've done interesting questions like,

54 00:02:15.570 --> 00:02:18.420 how does the structure of a network determine dynamics

55 00:02:18.420 --> 00:02:20.880 of processes on that network?

56 00:02:20.880 --> 00:02:23.700 And in turn, how do processes on that network

57 00:02:23.700 --> 00:02:25.443 give rise to structure?

58 00:02:29.580 --> 00:02:31.560 On the biological side,

59 00:02:31.560 --> 00:02:34.350 today's talk I'm going to be focusing on

60 00:02:34.350 --> 00:02:36.330 sort of applications of this question

61 00:02:36.330 --> 00:02:37.680 within neural networks.

62 00:02:37.680 --> 00:02:39.060 And I think that this sort of world of

63 00:02:39.060 --> 00:02:40.860 computational neuroscience is really exciting

64 00:02:40.860 --> 00:02:42.150 if you're interested in this interplay

65 00:02:42.150 --> 00:02:43.920 between structure and dynamics

66 00:02:43.920 --> 00:02:45.960 because neural networks encode, transmit

67 00:02:45.960 --> 00:02:49.140 and process information via dynamical processes.

68 00:02:49.140 --> 00:02:53.340 For example, the process, the dynamical process

69 00:02:53.340 --> 00:02:56.160 of a neural network is directed by the wiring patterns

70 00:02:56.160 --> 00:02:57.720 by the structure of that network.

71 00:02:57.720 --> 00:02:59.520 And moreover, if you're talking

72 00:02:59.520 --> 00:03:00.870 about some sort of learning process,

73 00:03:00.870 --> 00:03:02.520 then those wiring patterns can change

74 00:03:02.520 --> 00:03:04.860 and adapt during the learning process,

75 00:03:04.860 --> 00:03:06.423 so that are themselves dynamic.

76 00:03:07.800 --> 00:03:09.810 In this area I've been interested in questions like,

77 00:03:09.810 --> 00:03:11.760 how is the flow of information governed

78 00:03:11.760 --> 00:03:13.500 by the wiring pattern?

79 00:03:13.500 --> 00:03:16.230 How do patterns of information flow

80 00:03:16.230 --> 00:03:17.250 present themselves in data?

81 00:03:17.250 --> 00:03:19.140 And can they be inferred from that data?

82 00:03:19.140 --> 00:03:20.730 And what types of wiring patterns

83 00:03:20.730 --> 00:03:22.323 might develop during learning?

84 00:03:23.910 --> 00:03:25.500 Answering questions of this type requires

85 00:03:25.500 --> 00:03:26.340 a couple of things.

86 00:03:26.340 --> 00:03:28.860 Sort of very big picture, it requires a language

87 00:03:28.860 --> 00:03:30.930 for describing structures and patterns.

88 00:03:30.930 --> 00:03:32.550 It requires having a dynamical process,

89 00:03:32.550 --> 00:03:35.040 some sort of model for the neural net,

90 00:03:35.040 --> 00:03:37.530 and it requires a generating model

91 00:03:37.530 --> 00:03:40.080 that generates initial structure

92 00:03:40.080 --> 00:03:42.330 and links the structure to dynamics.

93 00:03:42.330 --> 00:03:45.420 Alternatively, if we don't generate things using a model,

94 00:03:45.420 --> 00:03:47.460 if we have some sort of observable or data,

95 00:03:47.460 --> 00:03:49.020 then we can try to work in the other direction

96 00:03:49.020 --> 00:03:51.540 and go from dynamics to structure.

97 00:03:51.540 --> 00:03:52.650 Today during this talk,

98 00:03:52.650 --> 00:03:55.470 I'm gonna be focusing really on this first piece,

99 00:03:55.470 --> 00:03:57.480 a language for describing structures and patterns.

100 00:03:57.480 --> 00:04:00.210 And on the second piece on sort of an observable

101 00:04:00.210 --> 00:04:04.260 that I've been working on trying to compute to use,

102 00:04:04.260 --> 00:04:07.530 to try to connect these three points together.

103 00:04:07.530 --> 00:04:10.140 So to get started, a little bit of biology.

104 00:04:10.140 --> 00:04:11.880 Really I was inspired in this project

105 00:04:11.880 --> 00:04:14.490 by a paper from K.G. Mura.

106 00:04:14.490 --> 00:04:16.650 Here he was looking at a coupled oscillator model,

107 00:04:16.650 --> 00:04:19.770 this is a Kuramoto model for neural activity

108 00:04:19.770 --> 00:04:22.140 where the firing dynamics interact with the wiring.

109 00:04:22.140 --> 00:04:24.630 So the wiring in the couples,

110 00:04:24.630 --> 00:04:28.860 the oscillators would adapt on a slower time scale

111 00:04:28.860 --> 00:04:31.440 than the oscillators themselves.

112 00:04:31.440 --> 00:04:33.570 And that adaptation could represent

113 00:04:33.570 --> 00:04:35.970 different types of learning processes.

114 00:04:35.970 --> 00:04:39.133 For example, the fire-together wire-together rules

115 00:04:39.133 --> 00:04:40.560 or Hebbian learning,

116 00:04:40.560 --> 00:04:43.110 you can look at causal learning rules,

117 00:04:43.110 --> 00:04:44.610 or anti-Hebbian learning rules.

118 00:04:44.610 --> 00:04:48.240 And this is just an example I've run of this system.

119 00:04:48.240 --> 00:04:49.980 This system of OD is sort of interesting

120 00:04:49.980 --> 00:04:52.410 because it can generate all sorts of different patterns.

121 00:04:52.410 --> 00:04:53.910 You can see synchronized firing,

122 00:04:53.910 --> 00:04:55.110 you can see traveling waves,

123 00:04:55.110 --> 00:04:56.610 you can see chaos,

124 00:04:56.610 --> 00:04:59.280 these occur at different sort of critical boundaries.

125 00:04:59.280 --> 00:05:01.170 So you can see phase transitions

126 00:05:01.170 --> 00:05:03.570 when you have large collections of these oscillators.

127 00:05:03.570 --> 00:05:05.100 And depending on how they're coupled together,

128 00:05:05.100 --> 00:05:06.333 it behaves differently.

129 00:05:07.410 --> 00:05:09.270 In particular some of what's interesting here is

130 00:05:09.270 --> 00:05:13.350 that starting from some random seed topology,

131 00:05:13.350 --> 00:05:16.170 the dynamics play forward from that initial condition,

132 00:05:16.170 --> 00:05:19.290 and that random seed topology produces some ensemble of

133 00:05:19.290 --> 00:05:22.020 wiring patterns that are of themselves random.

134 00:05:22.020 --> 00:05:23.850 You can think about the ensemble of wiring patterns

135 00:05:23.850 --> 00:05:25.200 as being chaotic,

136 00:05:25.200 --> 00:05:28.083 sort of realizations of some random initialization.

137 00:05:29.460 --> 00:05:31.560 That said, you can also observe structures

138 00:05:31.560 --> 00:05:33.360 within the systems of coupled oscillators.

139 00:05:33.360 --> 00:05:35.670 So you could see large scale cyclic structures

140 00:05:35.670 --> 00:05:37.830 representing organized causal firing patterns

141 00:05:37.830 --> 00:05:39.840 in certain regimes.

142 00:05:39.840 --> 00:05:41.760 So this is a nice example where graph structure

143 00:05:41.760 --> 00:05:43.710 and learning parameters can determine dynamics,

144 00:05:43.710 --> 00:05:46.560 and in turn where those dynamics can determine structure.

145 00:05:48.030 --> 00:05:49.260 On the other side, you can also think

146 00:05:49.260 --> 00:05:52.060 about a data-driven side instead of a model-driven side.

147 00:05:53.460 --> 00:05:55.590 If we empirically observe sample trajectories

148 00:05:55.590 --> 00:05:57.720 of some observables, for example, neuron recordings,

149 00:05:57.720 --> 00:05:59.070 then we might hope to infer something

150 00:05:59.070 --> 00:06:01.370 about the connectivity that generates them.

151 00:06:01.370 --> 00:06:03.750 And so here instead of starting by posing a model,

152 00:06:03.750 --> 00:06:06.000 and then simulating it and studying how it behaves,

153 00:06:06.000 --> 00:06:09.900 we can instead study data or try to study structure in data.

154 00:06:09.900 --> 00:06:12.420 Often that data comes in the form of covariance matrices

155 00:06:12.420 --> 00:06:14.040 representing firing rates.

156 00:06:14.040 --> 00:06:16.830 And these covariance matrices maybe auto covariance matrices

157 00:06:16.830 --> 00:06:18.180 with some sort of time lag.

158 00:06:19.110 --> 00:06:21.660 Here there are a couple of standard structural approaches,

159 00:06:21.660 --> 00:06:24.540 so there is a motivic expansion approach.

160 00:06:24.540 --> 00:06:28.350 This was at least introduced by Brent Doiron's lab

161 00:06:28.350 --> 00:06:30.450 with his student, Gay Walker.

162 00:06:30.450 --> 00:06:33.600 Here the idea is that you define some graph motifs,

163 00:06:33.600 --> 00:06:35.730 and then you can study the dynamics

164 00:06:35.730 --> 00:06:37.530 in terms of those graph motifs.

165 00:06:37.530 --> 00:06:41.010 For example, if you build a power series in those motifs,

166 00:06:41.010 --> 00:06:43.770 then you can try to represent your covariance matrices

167 00:06:43.770 --> 00:06:45.060 in terms of that power series.

168 00:06:45.060 --> 00:06:45.960 And this is something we're gonna talk

169 00:06:45.960 --> 00:06:47.130 about quite a bit today.

170 00:06:47.130 --> 00:06:49.350 This is really part of why I was inspired by this work is,

171 00:06:49.350 --> 00:06:51.450 I had been working separately on the idea of

172 00:06:51.450 --> 00:06:52.650 looking at covariance matrices

173 00:06:52.650 --> 00:06:54.903 in terms of these power series expansions.

174 00:06:56.040 --> 00:06:59.160 This is also connected to topological data analysis,

175 00:06:59.160 --> 00:07:01.170 and this is where the conversations with Lek-Heng

176 00:07:01.170 --> 00:07:02.940 played a role in this work.

177 00:07:02.940 --> 00:07:06.690 Topological data analysis aims to construct graphs,

178 00:07:06.690 --> 00:07:08.460 representing dynamical sort of systems.

179 00:07:08.460 --> 00:07:10.920 For example, you might look at the dynamical similarity

180 00:07:10.920 --> 00:07:12.990 of firing patterns of certain neurons,

181 00:07:12.990 --> 00:07:16.743 and then tries to study the topology of those graphs.

182 00:07:17.730 --> 00:07:19.530 Again, this sort of leads to similar questions,

183 00:07:19.530 --> 00:07:21.120 but we can be a little bit more precise here

184 00:07:21.120 --> 00:07:22.570 for thinking in neuroscience.

185 00:07:23.580 --> 00:07:25.350 We can say more precisely, for example,

186 00:07:25.350 --> 00:07:28.590 how is information processing and transfer represented,

187 00:07:28.590 --> 00:07:31.650 both in these covariance matrices and the structures

188 00:07:31.650 --> 00:07:33.390 that we hope to extract from them.

189 00:07:33.390 --> 00:07:34.740 In particular, can we try

190 00:07:34.740 --> 00:07:37.893 and infer causality from firing patterns?

191 00:07:39.420 --> 00:07:42.180 And this is fundamentally an information theoretic question.

192 00:07:42.180 --> 00:07:43.350 Really we're asking, can we study

193 00:07:43.350 --> 00:07:47.400 the directed exchange of information from trajectories?

194 00:07:47.400 --> 00:07:49.320 Here one approach, I mean, in some sense

195 00:07:49.320 --> 00:07:52.740 you can never tell causality without some underlying model,

196 00:07:52.740 --> 00:07:55.770 without some underlying understanding of the mechanism.

197 00:07:55.770 --> 00:07:57.540 So if all we can do is observe,

198 00:07:57.540 --> 00:08:00.510 then we need to define what we mean by causality.

199 00:08:00.510 --> 00:08:02.670 A reasonable sort of standard definition here

200 00:08:02.670 --> 00:08:03.780 is Wiener Causality,

201 00:08:03.780 --> 00:08:06.180 which says that two times series share a causal relation,

202 00:08:06.180 --> 00:08:08.040 so we say X causes Y,

203 00:08:08.040 --> 00:08:11.670 if the history of X informs a future of Y.

204 00:08:11.670 --> 00:08:14.250 Note that here ”cause” put in quotes,

205 00:08:14.250 --> 00:08:15.450 really means forecasts.

206 00:08:15.450 --> 00:08:18.180 That means that the past or the present of X

207 00:08:18.180 --> 00:08:21.630 carries relevant information about the future of Y.

208 00:08:21.630 --> 00:08:26.190 A natural measure of that information is transfer entropy.

209 00:08:26.190 --> 00:08:29.662 Transfer entropy was introduced by Schreiber in 2000,

210 00:08:29.662 --> 00:08:31.530 and it’s the expected KL divergence

211 00:08:31.530 --> 00:08:35.340 between the distribution of the future of Y

212 00:08:35.340 --> 00:08:38.010 given the history of X

213 00:08:38.010 --> 00:08:41.130 and the marginal distribution of the future of Y.

214 00:08:41.130 --> 00:08:43.110 So essentially it’s how much predictive information

215 00:08:43.110 --> 00:08:44.763 does X carry about Y?

216 00:08:46.080 --> 00:08:48.450 This is a nice quantity for a couple of reasons.

217 00:08:48.450 --> 00:08:51.330 First, it’s zero when two trajectories are independent.

218 00:08:51.330 --> 00:08:53.280 Second, since it’s just defined in terms of

219 00:08:53.280 --> 00:08:55.500 these conditional distributions, it’s model free.

220 00:08:55.500 --> 00:08:58.500 So I put here no with a star because this,

221 00:08:58.500 --> 00:09:00.660 the generative assumptions actually do matter

222 00:09:00.660 --> 00:09:01.650 when you go to try and compute it,

223 00:09:01.650 --> 00:09:04.530 but in principle it’s defined independent of the model.

224 00:09:04.530 --> 00:09:07.470 Again, unlike some other effective causality measures,

225 00:09:07.470 --> 00:09:11.340 it doesn’t require introducing some time lag to define.

226 00:09:11.340 --> 00:09:13.350 It’s a naturally directed quantity, right?

227 00:09:13.350 --> 00:09:14.640 We can say that the future of Y

228 00:09:14.640 --> 00:09:16.680 conditioned on the past of X and...

229 00:09:16.680 --> 00:09:19.590 That transfer entropy is defined on the terms of

230 00:09:19.590 --> 00:09:22.830 the future of Y conditioned on the past of X and Y.

231 00:09:22.830 --> 00:09:27.090 And that quantity is directed because reversing X and Y,

232 00:09:27.090 --> 00:09:29.670 it does not sort of symmetrically change this statement.

233 00:09:29.670 --> 00:09:30.930 This is different than quantities

234 00:09:30.930 --> 00:09:32.490 like mutual information or correlation

235 00:09:32.490 --> 00:09:34.290 that are also often used

236 00:09:34.290 --> 00:09:36.870 to try to measure effective connectivity in networks,

237 00:09:36.870 --> 00:09:39.843 which are fundamentally symmetric quantities.

238 00:09:41.400 --> 00:09:42.960 Transfer entropy is also less corrupted

239 00:09:42.960 --> 00:09:45.840 by measurement noise, linear mixing of signals,

240 00:09:45.840 --> 00:09:48.393 or shared coupling to external sources.

241 00:09:49.800 --> 00:09:51.870 Lastly, and maybe most interestingly,

242 00:09:51.870 --> 00:09:54.060 if we think in terms of correlations,

243 00:09:54.060 --> 00:09:55.590 correlations are really moments,

244 00:09:55.590 --> 00:09:57.360 correlations are really about covariances, right?

245 00:09:57.360 --> 00:09:58.980 Second order moments.

246 00:09:58.980 --> 00:10:00.810 Transfer entropies, these are about entropies,

247 00:10:00.810 --> 00:10:03.780 these are sort of logs of distributions,

248 00:10:03.780 --> 00:10:06.360 and so they depend on the full shape of these distributions,

249 00:10:06.360 --> 00:10:09.870 which means that transfer entropy can capture coupling

250 00:10:09.870 --> 00:10:13.080 that is maybe not apparent or not obvious,

251 00:10:13.080 --> 00:10:16.203 just looking at a second order moment type analysis.

252 00:10:17.280 --> 00:10:20.070 So transfer entropy has been applied pretty broadly.

253 00:10:20.070 --> 00:10:22.440 It's been applied to spiking cortical networks

254 00:10:22.440 --> 00:10:23.610 and calcium imaging,

255 00:10:23.610 --> 00:10:28.560 to MEG data in motor tasks and auditory discrimination.

256 00:10:28.560 --> 00:10:30.570 It's been applied to motion recognition,

257 00:10:30.570 --> 00:10:31.710 precious metal prices

258 00:10:31.710 --> 00:10:34.050 and multivariate time series forecasting,

259 00:10:34.050 --> 00:10:36.180 and more recently to accelerate learning

260 00:10:36.180 --> 00:10:38.040 in different artificial neural nets.

261 00:10:38.040 --> 00:10:39.990 So you can look at feedforward architectures,

262 00:10:39.990 --> 00:10:42.450 convolution architectures, even recurrent neural nets,

263 00:10:42.450 --> 00:10:43.830 and transfer entropy has been used

264 00:10:43.830 --> 00:10:46.443 to accelerate learning in those frameworks.

265 00:10:48.570 --> 00:10:49.590 For this part of the talk,

266 00:10:49.590 --> 00:10:52.470 I'd like to focus really on two questions.

267 00:10:52.470 --> 00:10:55.050 First, how do we compute transfer entropy?

268 00:10:55.050 --> 00:10:58.380 And then second, if we could compute transfer entropy

269 00:10:58.380 --> 00:10:59.700 and build a graph out of that,

270 00:10:59.700 --> 00:11:01.410 how would we study the structure of that graph?

271 00:11:01.410 --> 00:11:04.053 Essentially, how is information flow structured?

272 00:11:05.460 --> 00:11:07.810 We'll start with computing in transfer entropy.

273 00:11:09.120 --> 00:11:10.140 To compute transfer entropy,

274 00:11:10.140 --> 00:11:12.540 we actually need to write down an equation.

275 00:11:12.540 --> 00:11:14.400 So transfer entropy was originally introduced

276 00:11:14.400 --> 00:11:17.820 for discrete time arbitrary order Markov processes.

277 00:11:17.820 --> 00:11:20.520 So suppose we have two Markov processes: X and Y,

278 00:11:20.520 --> 00:11:22.920 and we'll let X of N denote the state of

279 00:11:22.920 --> 00:11:24.840 process X at time N,

280 00:11:24.840 --> 00:11:28.950 and XNK where the K is in superscript to denote the sequence

281 00:11:28.950 --> 00:11:32.010 starting from N minus K plus one going up to N.

282 00:11:32.010 --> 00:11:34.920 So that's sort of the last K states

283 00:11:34.920 --> 00:11:37.260 that the process X visited,

284 00:11:37.260 --> 00:11:39.990 then the transfer entropy from Y to X,

285 00:11:39.990 --> 00:11:42.580 they're denoted T, Y arrow to X

286 00:11:45.147 --> 00:11:50.130 is the entropy of the future of X conditioned on its past

287 00:11:50.130 --> 00:11:53.640 minus the entropy of the future of X conditioned on its past

288 00:11:53.640 --> 00:11:56.280 and the past of the trajectory Y.

289 00:11:56.280 --> 00:11:57.320 So here you can think the transfer entropy

290 00:11:57.320 --> 00:11:58.950 is essentially the reduction in entropy

291 00:11:58.950 --> 00:12:00.390 of the future states of X

292 00:12:00.390 --> 00:12:03.450 when incorporating the past of Y.

293 00:12:03.450 --> 00:12:04.950 This means that computing transfer entropy

294 00:12:04.950 --> 00:12:07.140 reduces to estimating essentially these entropies.

295 00:12:07.140 --> 00:12:08.490 That means we need to be able to estimate

296 00:12:08.490 --> 00:12:10.410 essentially the conditional distributions

297 00:12:10.410 --> 00:12:12.633 inside of these parentheses.

298 00:12:13.620 --> 00:12:15.390 That's easy for certain processes.

299 00:12:15.390 --> 00:12:18.660 So for example, if X and Y are Gaussian processes,

300 00:12:18.660 --> 00:12:20.160 then really what we're trying to compute

301 00:12:20.160 --> 00:12:21.690 is conditional mutual information.

302 00:12:21.690 --> 00:12:22.800 And there are nice equations

303 00:12:22.800 --> 00:12:24.510 for conditional mutual information

304 00:12:24.510 --> 00:12:26.220 when you have Gaussian random variables.

305 00:12:26.220 --> 00:12:29.250 So if I have three Gaussian random variables: X, Y, Z,

306 00:12:29.250 --> 00:12:32.700 possibly multivariate with joint covariance sigma,

307 00:12:32.700 --> 00:12:34.560 then the conditional mutual information

308 00:12:34.560 --> 00:12:35.670 between these variables,

309 00:12:35.670 --> 00:12:38.910 so the mutual information between X and Y conditioned on Z

310 00:12:38.910 --> 00:12:41.610 is just given by this ratio of log determinants

311 00:12:41.610 --> 00:12:42.663 of those covariances.

312 00:12:44.970 --> 00:12:48.210 In particular, a common test model used

313 00:12:48.210 --> 00:12:50.520 in sort of the transfer entropy literature

314 00:12:50.520 --> 00:12:52.530 are linear auto-regressive processes

315 00:12:52.530 --> 00:12:54.600 because a linear auto-regressive process

316 00:12:54.600 --> 00:12:56.550 when perturbed by Gaussian noise

317 00:12:56.550 --> 00:12:58.200 produces a Gaussian process.

318 00:12:58.200 --> 00:12:59.100 All of the different

319 00:12:59.100 --> 00:13:01.770 joint marginal conditional distributions are all Gaussian,

320 00:13:01.770 --> 00:13:03.090 which means that we can compute

321 00:13:03.090 --> 00:13:05.010 these covariances analytically,

322 00:13:05.010 --> 00:13:05.907 which then means that you can compute

323 00:13:05.907 --> 00:13:07.290 the transfer entropy analytically.

324 00:13:07.290 --> 00:13:08.940 So these linear auto-regressive processes

325 00:13:08.940 --> 00:13:10.080 are nice test cases

326 00:13:10.080 --> 00:13:12.450 because you can do everything analytically.

327 00:13:12.450 --> 00:13:14.880 They're also somewhat disappointing or somewhat limiting

328 00:13:14.880 --> 00:13:17.340 because in this linear auto-regressive case,

329 00:13:17.340 --> 00:13:20.223 transfer entropy is the same as Granger causality.

330 00:13:21.630 --> 00:13:23.910 And in this Gaussian case,

331 00:13:23.910 --> 00:13:25.920 essentially what we've done is we've reduced

332 00:13:25.920 --> 00:13:28.530 transfer entropy to a study of time-lagged correlations,

333 00:13:28.530 --> 00:13:29.640 so this becomes the same

334 00:13:29.640 --> 00:13:31.530 as sort of a correlation based analysis,

335 00:13:31.530 --> 00:13:34.350 we can't incorporate information beyond the second moments,

336 00:13:34.350 --> 00:13:36.390 if we restrict ourselves to Gaussian processes

337 00:13:36.390 --> 00:13:38.520 or Gaussian approximations.

338 00:13:38.520 --> 00:13:41.130 The other thing to note is this is strongly model-dependent

339 00:13:41.130 --> 00:13:42.630 because this particular formula

340 00:13:42.630 --> 00:13:43.890 for computing mutual information

341 00:13:43.890 --> 00:13:46.383 depends on having Gaussian distributions.

342 00:13:49.647 --> 00:13:53.220 In a more general setting or a more empirical setting,

343 00:13:53.220 --> 00:13:54.960 you might observe some data.

344 00:13:54.960 --> 00:13:56.130 You don't know if that data

345 00:13:56.130 --> 00:13:58.020 comes from some particular process,

346 00:13:58.020 --> 00:13:59.340 so you can't necessarily assume

347 00:13:59.340 --> 00:14:01.080 that conditional distributions are Gaussian,

348 00:14:01.080 --> 00:14:03.420 but we would still like to estimate transfer entropy,

349 00:14:03.420 --> 00:14:05.640 which leads to the problem of estimating transfer entropy

350 00:14:05.640 --> 00:14:08.040 given an observed time series.

351 00:14:08.040 --> 00:14:10.470 We would like to do this again, sans some model assumption,

352 00:14:10.470 --> 00:14:13.140 so we don't wanna assume Gaussianity.

353 00:14:13.140 --> 00:14:14.280 This is sort of trivial,

354 00:14:14.280 --> 00:14:16.920 again, I star that in discrete state spaces

355 00:14:16.920 --> 00:14:19.800 because essentially it amounts to counting occurrences,

356 00:14:19.800 --> 00:14:22.920 but it becomes difficult whenever the state spaces are large

357 00:14:22.920 --> 00:14:25.473 and/or high dimensional as they often are.

358 00:14:26.340 --> 00:14:28.440 This leads to a couple of different approaches.

359 00:14:28.440 --> 00:14:31.890 So as a first example, let's consider spike train data.

360 00:14:31.890 --> 00:14:34.890 So spike train data consists essentially of

361 00:14:34.890 --> 00:14:38.700 binning the state of a neuron into either on or off.

362 00:14:38.700 --> 00:14:41.460 So neurons, you can think either in the state zero or one,

363 00:14:41.460 --> 00:14:44.490 and then a pair wise calculation for transfer entropy

364 00:14:44.490 --> 00:14:47.640 only requires estimating a joint probability distribution

365 00:14:47.640 --> 00:14:50.910 over four to the K plus L states where K plus L,

366 00:14:50.910 --> 00:14:53.970 K is the history of X that we remember,

367 00:14:53.970 --> 00:14:55.713 and L is the history of Y.

368 00:14:57.430 --> 00:14:59.310 So if sort of the Markov process

369 00:14:59.310 --> 00:15:02.430 generating the spike train data is not of high order,

370 00:15:02.430 --> 00:15:04.200 does not have a long memory,

371 00:15:04.200 --> 00:15:06.390 then these K and L can be small,

372 00:15:06.390 --> 00:15:08.160 and this state space is fairly small,

373 00:15:08.160 --> 00:15:09.900 so this falls into that first category

374 00:15:09.900 --> 00:15:11.520 when we're looking at a discrete state space,

375 00:15:11.520 --> 00:15:13.023 and it's not too difficult.

376 00:15:14.880 --> 00:15:16.020 In a more general setting,

377 00:15:16.020 --> 00:15:17.640 if we don't try to bin the states

378 00:15:17.640 --> 00:15:19.380 of the neurons to on or off,

379 00:15:19.380 --> 00:15:22.110 for example, maybe we're looking at a firing rate model

380 00:15:22.110 --> 00:15:23.970 where we wanna look at the firing rates of the neurons,

381 00:15:23.970 --> 00:15:27.210 and that's a continuous random variable,

382 00:15:27.210 --> 00:15:29.250 then we need some other types of estimators.

383 00:15:29.250 --> 00:15:30.720 So the common estimator used here

384 00:15:30.720 --> 00:15:33.600 is a kernel density estimator, a KSG estimator,

385 00:15:33.600 --> 00:15:35.790 and this is designed for large continuous

386 00:15:35.790 --> 00:15:37.110 or high dimensional state spaces,

387 00:15:37.110 --> 00:15:39.273 e.g. sort of these firing rate models.

388 00:15:40.170 --> 00:15:43.320 Typically the approach is to employ a Takens delay map,

389 00:15:43.320 --> 00:15:45.120 which embeds your high dimensional data

390 00:15:45.120 --> 00:15:47.670 in some sort of lower dimensional space

391 00:15:47.670 --> 00:15:50.250 that tries to capture the intrinsic dimension

392 00:15:50.250 --> 00:15:54.600 of the attractor that your dynamic process settles onto.

393 00:15:54.600 --> 00:15:56.970 And then you try to estimate an unknown density

394 00:15:56.970 --> 00:15:59.430 based on this delay map using a k-nearest

395 00:15:59.430 --> 00:16:01.080 neighbor kernel density estimate.

396 00:16:01.080 --> 00:16:03.390 The advantage of this sort of

397 00:16:03.390 --> 00:16:04.593 k-nearest neighbor kernel density is

398 00:16:04.593 --> 00:16:07.440 that it dynamically adapts the width of the kernel,

399 00:16:07.440 --> 00:16:08.640 giving your sample density.

400 00:16:08.640 --> 00:16:11.310 And this has been implemented in some open source toolkits,

401 00:16:11.310 --> 00:16:13.673 these are the toolkits that we've been working with.

402 00:16:15.210 --> 00:16:17.640 So we've tested this in a couple of different models,

403 00:16:17.640 --> 00:16:18.780 and really I'd say this work,

404 00:16:18.780 --> 00:16:20.310 this is still very much work in progress,

405 00:16:20.310 --> 00:16:23.130 this is work that Bowen was developing over the summer.

406 00:16:23.130 --> 00:16:26.490 And so we developed a couple different models to test.

407 00:16:26.490 --> 00:16:29.310 The first were these Linear Auto-Regressive Networks,

408 00:16:29.310 --> 00:16:30.630 and we just used these to test

409 00:16:30.630 --> 00:16:31.800 the accuracy of the estimators

410 00:16:31.800 --> 00:16:33.270 because everything here is Gaussian,

411 00:16:33.270 --> 00:16:34.620 so you can compute the necessary

412 00:16:34.620 --> 00:16:36.900 transfer entropies analytically.

413 00:16:36.900 --> 00:16:38.820 The next more interesting class of networks

414 00:16:38.820 --> 00:16:41.520 are Threshold Linear Networks or TLNs.

415 00:16:41.520 --> 00:16:44.490 These are a firing rate model where your rate R

416 00:16:44.490 --> 00:16:46.590 obeys this sarcastic differential equation.

417 00:16:46.590 --> 00:16:50.940 So the rate of change and the rate, DR of T is,

418 00:16:50.940 --> 00:16:53.400 so you have sort of a leak term, negative RFT,

419 00:16:53.400 --> 00:16:56.940 and then plus here, this is essentially a coupling.

420 00:16:56.940 --> 00:17:00.330 All of this is inside here, the brackets with a plus,

421 00:17:00.330 --> 00:17:01.920 this is like a ReLU function,

422 00:17:01.920 --> 00:17:03.840 so this is just taking the positive part

423 00:17:03.840 --> 00:17:05.160 of what's on the inside.

424 00:17:05.160 --> 00:17:07.590 Here B is an activation threshold,

425 00:17:07.590 --> 00:17:09.060 W is a wiring matrix,

426 00:17:09.060 --> 00:17:10.860 and then R are those rates, again.

427 00:17:10.860 --> 00:17:13.200 And then C here, that's essentially covariance

428 00:17:13.200 --> 00:17:16.590 for some noise term for terming this process,

429 00:17:16.590 --> 00:17:19.260 we use these TLNs to test the sensitivity

430 00:17:19.260 --> 00:17:20.820 of our transfer entropy estimators

431 00:17:20.820 --> 00:17:23.730 to common and private noise sources as you change C,

432 00:17:23.730 --> 00:17:26.460 as well as sort of how well the entropy network

433 00:17:26.460 --> 00:17:29.433 agrees with the wiring matrix.

434 00:17:30.720 --> 00:17:32.490 A particular class of TLNs

435 00:17:32.490 --> 00:17:34.620 were really nice for these experiments

436 00:17:34.620 --> 00:17:36.990 what are called Combinatorial Threshold Linear Networks.

437 00:17:36.990 --> 00:17:38.070 These are really pretty new,

438 00:17:38.070 --> 00:17:42.270 these were introduced by Carina Curto's lab this year,

439 00:17:42.270 --> 00:17:45.240 and really this was inspired by a talk

440 00:17:45.240 --> 00:17:49.110 I'd seen her give at FACM in May.

441 00:17:49.110 --> 00:17:50.820 These are threshold linear networks

442 00:17:50.820 --> 00:17:52.320 where the weight matrix here, W,

443 00:17:52.320 --> 00:17:55.440 representing the wiring of the neurons

444 00:17:55.440 --> 00:17:58.020 is determined by a directed graph G.

445 00:17:58.020 --> 00:17:59.610 So you start with some directed graph G,

446 00:17:59.610 --> 00:18:00.810 that's what's shown here on the left.

447 00:18:00.810 --> 00:18:02.910 This figure is adapted from Carina's paper,

448 00:18:02.910 --> 00:18:03.743 this is a very nice paper

449 00:18:03.743 --> 00:18:05.470 if you'd like to take a look at it.

450 00:18:06.690 --> 00:18:09.003 And if I and J are not connected,

451 00:18:10.020 --> 00:18:12.030 then the weight matrix is assigned one value;

452 00:18:12.030 --> 00:18:14.460 and if they are connected, then it's assigned another value,

453 00:18:14.460 --> 00:18:18.300 and the wiring is zero if I equals J.

454 00:18:18.300 --> 00:18:19.710 These networks are nice

455 00:18:19.710 --> 00:18:21.930 if we wanna test structural hypotheses

456 00:18:21.930 --> 00:18:25.410 because it's very easy to predict from the input graph

457 00:18:25.410 --> 00:18:28.050 how the output dynamics of the network should behave,

458 00:18:28.050 --> 00:18:29.610 and they are really beautiful analysis

459 00:18:29.610 --> 00:18:31.530 that Carina does in this paper to show

460 00:18:31.530 --> 00:18:32.940 that you can produce all these different

461 00:18:32.940 --> 00:18:34.890 interlocking patterns of limit cycles

462 00:18:34.890 --> 00:18:36.420 and multistable states,

463 00:18:36.420 --> 00:18:38.220 and chaos, and all these nice patterns,

464 00:18:38.220 --> 00:18:40.530 and you can design them by picking these nice

465 00:18:40.530 --> 00:18:42.723 sort of directed graphs.

466 00:18:43.890 --> 00:18:46.230 The last class of networks that we've built to test

467 00:18:46.230 --> 00:18:47.760 are Leaky-Integrate and Fire Networks.

468 00:18:47.760 --> 00:18:51.000 So here we're using a Leaky-Integrate and Fire model

469 00:18:51.000 --> 00:18:54.390 where our wiring matrix, W, is drawn randomly.

470 00:18:54.390 --> 00:18:57.060 It's block stochastic, which means

471 00:18:57.060 --> 00:18:59.820 that it's (indistinct) between blocks.

472 00:18:59.820 --> 00:19:02.010 And it's a balanced network,

473 00:19:02.010 --> 00:19:04.200 so we have excitatory and inhibitory neurons

474 00:19:04.200 --> 00:19:06.180 that talk to each other,

475 00:19:06.180 --> 00:19:09.210 and maintain a sort of a balance in the dynamics here.

476 00:19:09.210 --> 00:19:11.340 The hope is to pick a large enough scale network

477 00:19:11.340 --> 00:19:13.380 that we see properly chaotic dynamics

478 00:19:13.380 --> 00:19:15.480 using this Leaky-Integrate and Fire model.

479 00:19:17.340 --> 00:19:20.760 These tests have yielded fairly mixed results,

480 00:19:20.760 --> 00:19:23.610 so the simple tests behave sort of as expected.

481 00:19:23.610 --> 00:19:26.760 So the estimators that are used are biased,

482 00:19:26.760 --> 00:19:28.560 and the bias typically decays slower

483 00:19:28.560 --> 00:19:30.030 than the variance in estimation,

484 00:19:30.030 --> 00:19:32.490 which means that you do need fairly long trajectories

485 00:19:32.490 --> 00:19:36.240 to try to properly estimate the transfer entropy.

486 00:19:36.240 --> 00:19:38.430 That said, transfer entropy does correctly identify

487 00:19:38.430 --> 00:19:40.320 causal relationships and simple graphs,

488 00:19:40.320 --> 00:19:43.980 and transfer entropy matches the underlying structure

489 00:19:43.980 --> 00:19:48.600 used in a Combinatorial Threshold Linear Network, so CTLN.

490 00:19:48.600 --> 00:19:52.200 Unfortunately, these results did not carry over as cleanly

491 00:19:52.200 --> 00:19:54.180 to the Leaky-Integrate and Fire models

492 00:19:54.180 --> 00:19:56.070 or to model sort of larger models.

493 00:19:56.070 --> 00:19:58.410 So what I'm showing you on the right here,

494 00:19:58.410 --> 00:20:00.240 this is a matrix where we've calculated

495 00:20:00.240 --> 00:20:03.150 the pairwise transfer entropy between all neurons

496 00:20:03.150 --> 00:20:06.240 in a 150 neuron balanced network.

497 00:20:06.240 --> 00:20:09.390 This has shown absolute, this has shown in the log scale.

498 00:20:09.390 --> 00:20:11.190 And the main thing I wanna highlight for it

499 00:20:11.190 --> 00:20:12.390 to taking a look at this matrix

500 00:20:12.390 --> 00:20:15.030 is very hard to see exactly what the structure is.

501 00:20:15.030 --> 00:20:16.530 You see this banding,

502 00:20:16.530 --> 00:20:19.830 that's because neurons tend to be highly predictive

503 00:20:19.830 --> 00:20:20.790 if they fire a lot.

504 00:20:20.790 --> 00:20:22.020 So there's a strong correlation

505 00:20:22.020 --> 00:20:25.410 between the transfer entropy, between X and Y,

506 00:20:25.410 --> 00:20:27.603 and just the activity level of X,

507 00:20:28.860 --> 00:20:31.170 but it's hard to distinguish block-wise differences,

508 00:20:31.170 --> 00:20:33.210 for example, between inhibitory neurons

509 00:20:33.210 --> 00:20:35.760 and excitatory neurons, and that really takes plotting out.

510 00:20:35.760 --> 00:20:38.640 So here this box in a whisker plot on the bottom,

511 00:20:38.640 --> 00:20:42.540 this is showing you if we group entries of this matrix

512 00:20:42.540 --> 00:20:43.530 by the type of connection,

513 00:20:43.530 --> 00:20:45.990 so maybe excitatory to excitatory,

514 00:20:45.990 --> 00:20:48.120 or inhibitory to excitatory, or so on,

515 00:20:48.120 --> 00:20:50.160 that the distribution of realized transfer entropy

516 00:20:50.160 --> 00:20:52.050 is really a different,

517 00:20:52.050 --> 00:20:54.120 but they're different in sort of subtle ways.

518 00:20:54.120 --> 00:20:57.273 So in this sort of larger scale balance network,

519 00:20:58.110 --> 00:21:02.370 it's much less clear whether transfer entropy

520 00:21:02.370 --> 00:21:05.160 effectively is like equated in some way

521 00:21:05.160 --> 00:21:07.803 with the true connectivity or wiring.

522 00:21:08.760 --> 00:21:10.230 In some ways, this is not a surprise

523 00:21:10.230 --> 00:21:11.760 because the behavior of the balance networks

524 00:21:11.760 --> 00:21:12.840 is inherently balanced,

525 00:21:12.840 --> 00:21:15.750 and (indistinct) inherently unstructured,

526 00:21:15.750 --> 00:21:18.330 but there are ways in which these experiments

527 00:21:18.330 --> 00:21:20.070 have sort of revealed confounding factors

528 00:21:20.070 --> 00:21:22.290 that are conceptual factors

529 00:21:22.290 --> 00:21:23.580 that make transfer entropies

530 00:21:23.580 --> 00:21:25.410 not as sort of an ideal measure,

531 00:21:25.410 --> 00:21:27.510 or maybe not as ideal as it seems

532 00:21:27.510 --> 00:21:29.400 given the start of this talk.

533 00:21:29.400 --> 00:21:32.850 So for example, suppose two trajectories:

534 00:21:32.850 --> 00:21:36.090 X and Y are both strongly driven by a third trajectory, Z,

535 00:21:36.090 --> 00:21:38.520 but X responds to Z first.

536 00:21:38.520 --> 00:21:40.380 Well, then the present information about X

537 00:21:40.380 --> 00:21:42.270 or the present state of X carries information

538 00:21:42.270 --> 00:21:45.000 about the future of Y, so X is predictive of Y,

539 00:21:45.000 --> 00:21:46.170 so X forecast Y.

540 00:21:46.170 --> 00:21:48.450 So in the transfer entropy or Wiener causality setting,

541 00:21:48.450 --> 00:21:50.790 we would say X causes Y,

542 00:21:50.790 --> 00:21:53.133 even if X and Y are only both responding to Z.

543 00:21:54.480 --> 00:21:55.980 So here in this example,

544 00:21:55.980 --> 00:21:58.560 suppose you have a directed tree where information

545 00:21:58.560 --> 00:22:02.100 or sort of dynamics propagate down the tree.

546 00:22:02.100 --> 00:22:06.570 If you look at this node here, PJ and I,

547 00:22:06.570 --> 00:22:08.460 PJ will react

548 00:22:08.460 --> 00:22:12.000 to essentially information traveling down this tree

549 00:22:12.000 --> 00:22:15.270 before I does, so PJ would be predictive for I.

550 00:22:15.270 --> 00:22:18.510 So we would observe an effective connection

551 00:22:18.510 --> 00:22:20.670 where PJ forecasts I,

552 00:22:20.670 --> 00:22:22.650 which means that neurons that are not directly connected

553 00:22:22.650 --> 00:22:24.420 may influence each other,

554 00:22:24.420 --> 00:22:25.920 and that this transfer entropy

555 00:22:25.920 --> 00:22:28.500 really you should think of in terms of forecasting,

556 00:22:28.500 --> 00:22:32.103 not in terms of being a direct analog to the wiring matrix.

557 00:22:33.270 --> 00:22:35.430 One way around this is to condition on the state

558 00:22:35.430 --> 00:22:36.870 of the rest of the network

559 00:22:36.870 --> 00:22:38.520 before you start doing some averaging.

560 00:22:38.520 --> 00:22:40.890 This leads to some other notions of entropy.

561 00:22:40.890 --> 00:22:42.450 So for example, causation entropy,

562 00:22:42.450 --> 00:22:43.800 and this is sort of a promising direction,

563 00:22:43.800 --> 00:22:45.993 but it's not a time to explore yet.

564 00:22:47.310 --> 00:22:49.260 So that's the estimation side,

565 00:22:49.260 --> 00:22:51.630 those are the tools for estimating transfer entropy.

566 00:22:51.630 --> 00:22:52.800 Now let's switch gears

567 00:22:52.800 --> 00:22:55.170 and talk about that second question I had introduced,

568 00:22:55.170 --> 00:22:57.450 which is essentially, how do we analyze structure?

569 00:22:57.450 --> 00:23:00.450 Suppose we could calculate a transfer entropy graph,

570 00:23:00.450 --> 00:23:03.600 how would we extract structural information from that graph?

571 00:23:03.600 --> 00:23:06.240 And here, I'm going to be introducing some tools

572 00:23:06.240 --> 00:23:07.530 that I've worked on for awhile

573 00:23:07.530 --> 00:23:11.370 for describing sort of random structures and graphs.

574 00:23:11.370 --> 00:23:14.700 These are tied back to some work I'd really done

575 00:23:14.700 --> 00:23:17.730 as a graduate student in conversations with Lek-Heng.

576 00:23:17.730 --> 00:23:19.290 So we start in a really simple context,

577 00:23:19.290 --> 00:23:20.670 which is the graph or network.

578 00:23:20.670 --> 00:23:22.380 This could be directed or undirected,

579 00:23:22.380 --> 00:23:24.360 however, we're gonna require that does not have self-loops,

580 00:23:24.360 --> 00:23:25.650 then it's finite.

581 00:23:25.650 --> 00:23:27.930 We'll let V here be the number of vertices

582 00:23:27.930 --> 00:23:30.390 and E be the number of edges.

583 00:23:30.390 --> 00:23:32.730 Then the object of study that we'll introduce

584 00:23:32.730 --> 00:23:34.020 is something called an edge flow.

585 00:23:34.020 --> 00:23:35.340 An edge flow is essentially a function

586 00:23:35.340 --> 00:23:36.810 on the edges of the graph.

587 00:23:36.810 --> 00:23:39.870 So this is a function that accepts pairs of endpoints

588 00:23:39.870 --> 00:23:41.580 and returns a real number,

589 00:23:41.580 --> 00:23:42.990 and this is an alternating function.

590 00:23:42.990 --> 00:23:44.880 So if I had to take F of IJ,

591 00:23:44.880 --> 00:23:46.710 that's negative F of JI

592 00:23:46.710 --> 00:23:49.350 because you can think of F of IJ as being some flow,

593 00:23:49.350 --> 00:23:51.810 like a flow of material between I and J,

594 00:23:51.810 --> 00:23:53.910 hence this name, edge flow.

595 00:23:53.910 --> 00:23:55.620 This is analogous to a vector field

596 00:23:55.620 --> 00:23:57.510 because this is like the analogous construction

597 00:23:57.510 --> 00:23:58.890 to a vector field in the graph,

598 00:23:58.890 --> 00:24:01.950 and represents some sort of flow between nodes.

599 00:24:01.950 --> 00:24:04.440 Edge flows are really sort of generic things,

600 00:24:04.440 --> 00:24:06.900 so you can take this idea of an edge flow

601 00:24:06.900 --> 00:24:08.910 and apply it in a lot of different areas

602 00:24:08.910 --> 00:24:09.990 because really all you need is,

603 00:24:09.990 --> 00:24:11.970 you just need to structure some alternating function

604 00:24:11.970 --> 00:24:13.410 on the edges of the graph.

605 00:24:13.410 --> 00:24:16.140 So I've sort of read papers

606 00:24:16.140 --> 00:24:18.600 and worked in a bunch of these different areas,

607 00:24:18.600 --> 00:24:20.640 particularly I've focused on applications of this

608 00:24:20.640 --> 00:24:24.660 in game theory, in pairwise and social choice settings,

609 00:24:24.660 --> 00:24:26.130 in biology and Markov chains.

610 00:24:26.130 --> 00:24:28.170 And a lot of this project has been attempting

611 00:24:28.170 --> 00:24:31.320 to take this experience working with edge flows in,

612 00:24:31.320 --> 00:24:34.140 for example, say non-equilibrium thermodynamics

613 00:24:34.140 --> 00:24:35.940 or looking at pairwise preference data,

614 00:24:35.940 --> 00:24:37.830 and looking at a different application area

615 00:24:37.830 --> 00:24:39.630 here to neuroscience.

616 00:24:39.630 --> 00:24:41.580 Really you could think about the edge flow

617 00:24:41.580 --> 00:24:43.170 or a relevant edge flow in neuroscience,

618 00:24:43.170 --> 00:24:45.780 you might be asking about asymmetries and wiring patterns,

619 00:24:45.780 --> 00:24:48.840 or differences in directed influence or causality,

620 00:24:48.840 --> 00:24:50.280 or really you could think about these

621 00:24:50.280 --> 00:24:51.270 transfer entropy quantities.

622 00:24:51.270 --> 00:24:53.010 This is why I was excited about transfer entropy.

623 00:24:53.010 --> 00:24:55.770 Transfer entropy is inherently directed notion

624 00:24:55.770 --> 00:24:57.390 of information flow,

625 00:24:57.390 --> 00:24:58.560 so it's natural to think

626 00:24:58.560 --> 00:25:01.380 that if you can calculate things like a transfer entropy,

627 00:25:01.380 --> 00:25:02.520 then really what you're studying

628 00:25:02.520 --> 00:25:04.370 is some sort of edge flow on a graph.

629 00:25:05.820 --> 00:25:08.340 Edge flows often are subject to

630 00:25:08.340 --> 00:25:10.200 sort of the same set of common questions.

631 00:25:10.200 --> 00:25:12.150 So if I wanna analyze the structure of an edge flow,

632 00:25:12.150 --> 00:25:13.770 there's some really big global questions

633 00:25:13.770 --> 00:25:15.120 that I would often ask,

634 00:25:15.120 --> 00:25:17.920 that get asked in all these different application areas.

635 00:25:19.140 --> 00:25:20.340 One common question is,

636 00:25:20.340 --> 00:25:22.710 well, does the flow originate somewhere and end somewhere?

637 00:25:22.710 --> 00:25:25.020 Are there sources and sinks in the graph?

638 00:25:25.020 --> 00:25:26.067 Another is, does it circulate?

639 00:25:26.067 --> 00:25:29.073 And if it does circulate, on what scales and where?

640 00:25:30.720 --> 00:25:32.520 If you have a network that's connected

641 00:25:32.520 --> 00:25:34.410 to a whole exterior network,

642 00:25:34.410 --> 00:25:36.540 for example, if you're looking at some small subsystem

643 00:25:36.540 --> 00:25:38.310 that's embedded in a much larger system

644 00:25:38.310 --> 00:25:40.710 as is almost always the case in neuroscience,

645 00:25:40.710 --> 00:25:42.000 then you also need to think about,

646 00:25:42.000 --> 00:25:43.290 what passes through the network?

647 00:25:43.290 --> 00:25:45.540 So is there a flow or a current that moves

648 00:25:45.540 --> 00:25:46.980 through the boundary of the network?

649 00:25:46.980 --> 00:25:50.520 Is there information that flows through the network

650 00:25:50.520 --> 00:25:52.230 that you're studying?

651 00:25:52.230 --> 00:25:54.660 And in particular if we have these different types of flow,

652 00:25:54.660 --> 00:25:56.640 if flow can originate and source and end in sinks,

653 00:25:56.640 --> 00:25:59.040 if it can circulate, if it can pass through,

654 00:25:59.040 --> 00:26:02.550 can we decompose the flow into pieces that do each of these,

655 00:26:02.550 --> 00:26:05.200 and ask how much of the flow does one, two, or three?

656 00:26:06.810 --> 00:26:09.333 Those questions lead to a decomposition.

657 00:26:10.590 --> 00:26:13.470 So here we're going to start with this simple idea,

658 00:26:13.470 --> 00:26:14.940 we're going to decompose an edge flow

659 00:26:14.940 --> 00:26:17.430 by projecting it onto orthogonal subspaces

660 00:26:17.430 --> 00:26:20.040 associated with some graph operators.

661 00:26:20.040 --> 00:26:24.030 Generically if we consider two linear operators: A and B,

662 00:26:24.030 --> 00:26:26.760 where the product A times B equals zero,

663 00:26:26.760 --> 00:26:29.160 then the range of B must be contained

664 00:26:29.160 --> 00:26:31.350 in the null space of A,

665 00:26:31.350 --> 00:26:33.420 which means that I can express

666 00:26:33.420 --> 00:26:34.950 essentially any set of real numbers.

667 00:26:34.950 --> 00:26:37.500 So you can think of this as being the vector space

668 00:26:37.500 --> 00:26:42.500 of possible edge flows as a direct sum of the range of B,

669 00:26:42.690 --> 00:26:44.730 the range of A transpose

670 00:26:44.730 --> 00:26:47.250 and the intersection of the null space of B transpose

671 00:26:47.250 --> 00:26:48.420 in the null space of A.

672 00:26:48.420 --> 00:26:52.680 This blue subspace, this is called the harmonic space,

673 00:26:52.680 --> 00:26:54.100 and this is trivial

674 00:26:55.620 --> 00:26:57.810 in many applications

675 00:26:57.810 --> 00:26:59.790 if you choose A and B correctly.

676 00:26:59.790 --> 00:27:02.220 So there's often settings where you can pick A and B,

677 00:27:02.220 --> 00:27:05.700 so that these two null spaces have no intersection,

678 00:27:05.700 --> 00:27:07.860 and then this decomposition boils down

679 00:27:07.860 --> 00:27:10.350 to just separating a vector space

680 00:27:10.350 --> 00:27:14.373 into the range of B and the range of A transpose.

681 00:27:15.780 --> 00:27:16.980 In the graph setting,

682 00:27:16.980 --> 00:27:19.260 our goal is essentially to pick these operators

683 00:27:19.260 --> 00:27:20.430 to the meaningful things.

684 00:27:20.430 --> 00:27:21.900 That is to pick graph operators,

685 00:27:21.900 --> 00:27:25.890 so that these subspaces carry a meaningful,

686 00:27:25.890 --> 00:27:29.700 or carry meaning in the structural context.

687 00:27:29.700 --> 00:27:33.480 So let's think a little bit about graph operators here,

688 00:27:33.480 --> 00:27:35.490 so let's look at two different classes of operators.

689 00:27:35.490 --> 00:27:40.350 So we can consider matrices that have E rows and N columns,

690 00:27:40.350 --> 00:27:43.500 or matrices that have L rows and E columns where,

691 00:27:43.500 --> 00:27:45.800 again, E is the number of edges in this graph.

692 00:27:47.790 --> 00:27:50.190 If I have a matrix with E rows,

693 00:27:50.190 --> 00:27:53.370 then each column of the matrix has as many entries

694 00:27:53.370 --> 00:27:54.960 as there are edges in the graph,

695 00:27:54.960 --> 00:27:57.420 so it can be thought of as itself an edge flow.

696 00:27:57.420 --> 00:27:59.250 So you could think that this matrix is composed

697 00:27:59.250 --> 00:28:01.620 of a set of columns where each column is some particular

698 00:28:01.620 --> 00:28:04.173 sort of motivic flow or flow motif.

699 00:28:05.430 --> 00:28:09.450 In contrast if I look at a matrix where I have E columns,

700 00:28:09.450 --> 00:28:11.430 then each row of the matrix is a flow motif,

701 00:28:11.430 --> 00:28:14.400 so products against M

702 00:28:14.400 --> 00:28:18.360 evaluate inner products against specific flow motifs.

703 00:28:18.360 --> 00:28:19.620 That means that in this context,

704 00:28:19.620 --> 00:28:21.090 if I look at the range of this matrix,

705 00:28:21.090 --> 00:28:22.710 this is really a linear combination

706 00:28:22.710 --> 00:28:25.230 of a specific subset of flow motifs.

707 00:28:25.230 --> 00:28:26.340 And in this context,

708 00:28:26.340 --> 00:28:27.780 if I look at the null space of the matrix,

709 00:28:27.780 --> 00:28:30.030 I'm looking at all edge flows orthogonal

710 00:28:30.030 --> 00:28:32.040 to that set of flow motifs.

711 00:28:32.040 --> 00:28:36.240 So here if I look at the range of a matrix with E rows,

712 00:28:36.240 --> 00:28:38.730 that subspace is essentially a modeling behavior

713 00:28:38.730 --> 00:28:40.170 similar to the motifs.

714 00:28:40.170 --> 00:28:43.680 So if I pick a set of motifs that flow out of a node

715 00:28:43.680 --> 00:28:45.180 or flow into a node,

716 00:28:45.180 --> 00:28:48.180 then this range is going to be a subspace of edge flows

717 00:28:48.180 --> 00:28:51.330 that tend to originate in sources and end in sinks.

718 00:28:51.330 --> 00:28:53.790 In contrast here, the null space of M,

719 00:28:53.790 --> 00:28:56.910 that's all edge flows orthogonal to the flow motifs,

720 00:28:56.910 --> 00:28:59.010 so it models behavior distinct from the motifs.

721 00:28:59.010 --> 00:29:02.490 Essentially this space asks, what doesn't the flow do?

722 00:29:02.490 --> 00:29:04.840 Whereas this space asks, what does the flow do?

723 00:29:06.540 --> 00:29:09.180 Here is a simple, sort of very classical example.

724 00:29:09.180 --> 00:29:10.710 And really this goes all the way back to,

725 00:29:10.710 --> 00:29:13.710 you could think like Kirchhoff electric circuit theory.

726 00:29:13.710 --> 00:29:15.180 We can define two operators.

727 00:29:15.180 --> 00:29:17.850 Here G, this is essentially a gradient operator.

728 00:29:17.850 --> 00:29:19.830 And if you've taken some graph theory,

729 00:29:19.830 --> 00:29:22.320 you might know this as the edge incidence matrix.

730 00:29:22.320 --> 00:29:24.930 This is a matrix which essentially records

731 00:29:24.930 --> 00:29:26.400 the endpoints of an edge

732 00:29:26.400 --> 00:29:29.100 and evaluates differences across it.

733 00:29:29.100 --> 00:29:32.760 So, for example, if I look at this first row of G,

734 00:29:32.760 --> 00:29:35.340 this corresponds to edge one in the graph,

735 00:29:35.340 --> 00:29:38.670 and if I had a function defined on the nodes in the graph,

736 00:29:38.670 --> 00:29:42.780 products with G would evaluate differences across this edge.

737 00:29:42.780 --> 00:29:44.340 If you look at its columns,

738 00:29:44.340 --> 00:29:45.930 each column here is a flow motif.

739 00:29:45.930 --> 00:29:48.900 So, for example, this highlighted second column,

740 00:29:48.900 --> 00:29:51.510 this is entries: one, negative one, zero, negative one.

741 00:29:51.510 --> 00:29:53.070 If you carry those back to the edges,

742 00:29:53.070 --> 00:29:56.100 that corresponds to this specific flow motif.

743 00:29:56.100 --> 00:29:57.810 So here this gradient,

744 00:29:57.810 --> 00:30:00.300 it's adjoint to essentially a divergence operator,

745 00:30:00.300 --> 00:30:03.300 which means that the flow motifs are unit inflows

746 00:30:03.300 --> 00:30:05.190 or unit outflows from specific nodes,

747 00:30:05.190 --> 00:30:07.170 like what's shown here.

748 00:30:07.170 --> 00:30:09.540 You can also introduce something like a curl operator.

749 00:30:09.540 --> 00:30:13.200 The curl operator evaluates paths, sums around loops.

750 00:30:13.200 --> 00:30:16.170 So this row here, for example, this is a flow motif

751 00:30:16.170 --> 00:30:20.430 corresponding to the loop labeled A in this graph.

752 00:30:20.430 --> 00:30:21.330 You could certainly imagine

753 00:30:21.330 --> 00:30:23.400 other operators' built cutter, other motifs,

754 00:30:23.400 --> 00:30:25.020 these operators are particularly nice

755 00:30:25.020 --> 00:30:27.070 because they define principled subspaces.

756 00:30:28.200 --> 00:30:30.990 So if we apply that generic decomposition,

757 00:30:30.990 --> 00:30:32.220 then we could say that the space

758 00:30:32.220 --> 00:30:34.080 of possible edge flows are E,

759 00:30:34.080 --> 00:30:37.410 it can be decomposed into the range of the grading operator,

760 00:30:37.410 --> 00:30:39.480 the range of the curl transpose,

761 00:30:39.480 --> 00:30:41.640 and the intersection of their null spaces

762 00:30:41.640 --> 00:30:43.770 into this harmonic space.

763 00:30:43.770 --> 00:30:46.340 This is nice because the range of the gradient that flows,

764 00:30:46.340 --> 00:30:47.730 it start and end somewhere.

765 00:30:47.730 --> 00:30:49.500 Those are flows that are associated with

766 00:30:49.500 --> 00:30:51.990 like motion down a potential.

767 00:30:51.990 --> 00:30:53.220 So these if you're thinking physics,

768 00:30:53.220 --> 00:30:54.630 you might say that these are sort of conservative,

769 00:30:54.630 --> 00:30:56.520 these are like flows generated by a voltage

770 00:30:56.520 --> 00:30:58.680 if you're looking at electric circuit.

771 00:30:58.680 --> 00:31:00.840 These cyclic flows, well, these are the flows

772 00:31:00.840 --> 00:31:02.730 in the range of the curl transpose,

773 00:31:02.730 --> 00:31:03.840 and then this harmonic space,

774 00:31:03.840 --> 00:31:06.360 those are flows that enter and leave the network

775 00:31:06.360 --> 00:31:08.940 without either starting or ending

776 00:31:08.940 --> 00:31:11.040 a sink or a source, or circulating.

777 00:31:11.040 --> 00:31:13.170 So you can think that really this decomposes

778 00:31:13.170 --> 00:31:15.540 the space of edge flows into flows that start

779 00:31:15.540 --> 00:31:17.220 and end somewhere inside the network.

780 00:31:17.220 --> 00:31:19.110 Flows that circulate within the network,

781 00:31:19.110 --> 00:31:20.310 and flows that do neither,

782 00:31:20.310 --> 00:31:22.470 i.e. flows that enter and leave the network.

783 00:31:22.470 --> 00:31:25.140 So this accomplishes that initial decomposition

784 00:31:25.140 --> 00:31:26.390 I'd set out at the start.

785 00:31:28.110 --> 00:31:31.320 Once we have this decomposition, then we can evaluate

786 00:31:31.320 --> 00:31:34.440 the sizes of the components of decomposition to measure

787 00:31:34.440 --> 00:31:37.500 how much of the flow starts and ends somewhere,

788 00:31:37.500 --> 00:31:39.300 how much circulates and so on.

789 00:31:39.300 --> 00:31:41.370 So we can introduce these generic measures

790 00:31:41.370 --> 00:31:44.100 we're given some operator N,

791 00:31:44.100 --> 00:31:45.960 we decompose the space of edge flows

792 00:31:45.960 --> 00:31:49.020 into the range of M and the null space of M transpose,

793 00:31:49.020 --> 00:31:52.050 which means we can project F onto these subspaces,

794 00:31:52.050 --> 00:31:54.570 and then just evaluate the sizes of these components.

795 00:31:54.570 --> 00:31:56.580 And that's a way of measuring

796 00:31:56.580 --> 00:31:58.530 how much of the flow behaves like

797 00:31:58.530 --> 00:32:00.630 the flow motifs contained in this operator,

798 00:32:00.630 --> 00:32:01.830 and how much it doesn't.

799 00:32:04.080 --> 00:32:06.690 So, yeah, so that lets us answer this question,

800 00:32:06.690 --> 00:32:08.760 and this is the tool that we're going to be using

801 00:32:08.760 --> 00:32:10.893 sort of as our measurable.

802 00:32:12.270 --> 00:32:15.510 Now that's totally easy to do,

803 00:32:15.510 --> 00:32:17.370 if you're given a fixed edge flow and a fixed graph

804 00:32:17.370 --> 00:32:18.330 because if you have fixed graph,

805 00:32:18.330 --> 00:32:20.460 you can build your operators, you choose the motifs,

806 00:32:20.460 --> 00:32:23.100 you have fixed edge flow, you just project the edge flow

807 00:32:23.100 --> 00:32:25.020 onto the subspaces spanned by those operators,

808 00:32:25.020 --> 00:32:25.853 and you're done.

809 00:32:26.910 --> 00:32:30.570 However, there are many cases where it's worth thinking

810 00:32:30.570 --> 00:32:32.850 about a distribution of edge flows,

811 00:32:32.850 --> 00:32:35.913 and then expected structures given that distribution.

812 00:32:36.780 --> 00:32:39.120 So here we're going to be considering random edge flows,

813 00:32:39.120 --> 00:32:40.740 for example, in edge flow capital F,

814 00:32:40.740 --> 00:32:43.350 here I'm using capital letters to denote random quantities

815 00:32:43.350 --> 00:32:44.940 sampled from an edge flow distributions.

816 00:32:44.940 --> 00:32:46.470 This is a distribution of possible edge flows.

817 00:32:46.470 --> 00:32:48.360 And this is worth thinking about

818 00:32:48.360 --> 00:32:51.480 because many generative models are stochastic.

819 00:32:51.480 --> 00:32:52.980 They may involve some random seed,

820 00:32:52.980 --> 00:32:54.870 or they may, for example, like that neural model

821 00:32:54.870 --> 00:32:57.780 or a lot of these sort of neural models be chaotic.

822 00:32:57.780 --> 00:33:01.050 So even if they are deterministic generative models,

823 00:33:01.050 --> 00:33:02.550 the output data behaves

824 00:33:02.550 --> 00:33:04.523 as it was sampled from the distribution.

825 00:33:05.430 --> 00:33:07.020 On the empirical side, for example,

826 00:33:07.020 --> 00:33:09.030 when we're estimating transfer entropy

827 00:33:09.030 --> 00:33:11.070 or estimating some information flow,

828 00:33:11.070 --> 00:33:13.380 then there's always some degree of measurement error

829 00:33:13.380 --> 00:33:15.420 or uncertainty in that estimate,

830 00:33:15.420 --> 00:33:17.520 which really means sort of from a Bayesian perspective,

831 00:33:17.520 --> 00:33:19.720 we should be thinking that our estimator

832 00:33:20.580 --> 00:33:22.650 is a point estimate drawn from some

833 00:33:22.650 --> 00:33:24.030 posterior distribution of edge flows,

834 00:33:24.030 --> 00:33:25.260 and then we're back in the setting where,

835 00:33:25.260 --> 00:33:27.780 again, we need to talk about a distribution.

836 00:33:27.780 --> 00:33:30.720 Lastly, this random edge flow setting is also

837 00:33:30.720 --> 00:33:33.640 really important if we wanna compare to null hypotheses

838 00:33:34.740 --> 00:33:36.990 because often if you want to compare

839 00:33:36.990 --> 00:33:38.370 to some sort of null hypothesis,

840 00:33:38.370 --> 00:33:40.920 it's helpful to have an ensemble of edge flows

841 00:33:40.920 --> 00:33:43.560 to compare against, which means that we would like

842 00:33:43.560 --> 00:33:45.510 to be able to talk about expected structure

843 00:33:45.510 --> 00:33:47.763 under varying distributional assumptions.

844 00:33:49.650 --> 00:33:54.150 If we can talk meaningfully about random edge flows,

845 00:33:54.150 --> 00:33:56.190 then really what we can start doing is

846 00:33:56.190 --> 00:33:58.920 we can start bridging the expected structure

847 00:33:58.920 --> 00:34:00.240 back to the distribution.

848 00:34:00.240 --> 00:34:03.000 So what we're looking for is a way of explaining

849 00:34:03.000 --> 00:34:04.620 sort of generic expectations

850 00:34:04.620 --> 00:34:06.990 of what the structure will look like

851 00:34:06.990 --> 00:34:09.690 as we vary this distribution of edge flows.

852 00:34:09.690 --> 00:34:12.720 You could think that a particular dynamical system

853 00:34:12.720 --> 00:34:16.530 generates a wiring pattern,

854 00:34:16.530 --> 00:34:19.260 that generates firing dynamics,

855 00:34:19.260 --> 00:34:20.730 those firing dynamics determine

856 00:34:20.730 --> 00:34:23.190 some sort of information flow graph.

857 00:34:23.190 --> 00:34:24.690 And then that information flow graph

858 00:34:24.690 --> 00:34:27.750 is really a sample from that generative model.

859 00:34:27.750 --> 00:34:30.480 And we would like to be able to talk about,

860 00:34:30.480 --> 00:34:32.760 what would we expect if we knew the distribution

861 00:34:32.760 --> 00:34:35.310 of edge flows about the global structure?

862 00:34:35.310 --> 00:34:36.960 That is, we'd like to bridge global structure

863 00:34:36.960 --> 00:34:38.670 back to this distribution,

864 00:34:38.670 --> 00:34:41.400 and then ideally you would bridge that distribution back

865 00:34:41.400 --> 00:34:42.420 to the generative mechanism.

866 00:34:42.420 --> 00:34:44.670 This is a project for a future work,

867 00:34:44.670 --> 00:34:46.650 obviously this is fairly ambitious.

868 00:34:46.650 --> 00:34:49.350 However, this first point is something that you can do

869 00:34:50.610 --> 00:34:53.040 really in fairly explicit detail.

870 00:34:53.040 --> 00:34:54.180 And that's what I'd like to spell out

871 00:34:54.180 --> 00:34:55.440 with the end of this talk is

872 00:34:55.440 --> 00:34:58.080 how do you bridge global structure

873 00:34:58.080 --> 00:34:59.943 back to a distribution of edge flows?

874 00:35:02.220 --> 00:35:04.500 So, yeah, so that's the main question,

875 00:35:04.500 --> 00:35:06.240 how does the choice of distribution

876 00:35:06.240 --> 00:35:08.553 influence the expected global flow structure?

877 00:35:12.000 --> 00:35:14.790 So first, we start with the Lemma.

878 00:35:14.790 --> 00:35:17.010 Suppose that we have a distribution of edge flows

879 00:35:17.010 --> 00:35:19.920 with some expectation F bar and some covariance,

880 00:35:19.920 --> 00:35:23.640 here I'm using double bar V to denote covariance.

881 00:35:23.640 --> 00:35:26.300 We'll let S contained in the set of,

882 00:35:26.300 --> 00:35:28.680 or S be a subspace

883 00:35:28.680 --> 00:35:31.110 contained within the vector space of edge flows,

884 00:35:31.110 --> 00:35:35.100 and we'll let Ps of S be the orthogonal projector onto S.

885 00:35:35.100 --> 00:35:40.100 Then Fs of S, that's the projection F onto this subspace S,

886 00:35:40.140 --> 00:35:42.900 the expectation of its norm squared

887 00:35:42.900 --> 00:35:47.900 is the norm of the expected flow projected onto S squared.

888 00:35:48.390 --> 00:35:51.760 So this is essentially the expectation of the sample

889 00:35:52.680 --> 00:35:55.800 is the measure evaluated of the expected sample.

890 00:35:55.800 --> 00:35:58.140 And then plus a term that involves an inner product

891 00:35:58.140 --> 00:36:00.240 between the projector onto the subspace,

892 00:36:00.240 --> 00:36:02.160 and the covariance matrix for the edge flows.

893 00:36:02.160 --> 00:36:03.960 Here this denotes the matrix inner product,

894 00:36:03.960 --> 00:36:06.993 so this is just the sum overall IJ entries.

895 00:36:09.030 --> 00:36:10.230 What's nice about this formula

896 00:36:10.230 --> 00:36:14.380 is at least in terms of expectation, it reduces the study

897 00:36:15.660 --> 00:36:18.210 of the bridge between distribution

898 00:36:18.210 --> 00:36:21.660 and network structure to a study of moments, right?

899 00:36:21.660 --> 00:36:23.520 Because we've replaced the distributional problem here

900 00:36:23.520 --> 00:36:26.730 with a linear algebra problem

901 00:36:26.730 --> 00:36:28.740 that's posed in terms of this projector,

902 00:36:28.740 --> 00:36:30.570 the projector under the subspace S,

903 00:36:30.570 --> 00:36:33.360 which is determined by the topology of the network,

904 00:36:33.360 --> 00:36:35.760 and the variance in that edge flow

905 00:36:35.760 --> 00:36:38.010 which is determined by your generative model.

906 00:36:39.660 --> 00:36:42.150 Well, you might say, okay, well, (laughs) fine,

907 00:36:42.150 --> 00:36:43.920 this is a matrix inner product, we can just stop here,

908 00:36:43.920 --> 00:36:45.000 we could compute this projector,

909 00:36:45.000 --> 00:36:47.010 we could sample a whole bunch of edge flows,

910 00:36:47.010 --> 00:36:47.843 compute this covariance.

911 00:36:47.843 --> 00:36:50.070 So you can do this matrix inner product,

912 00:36:50.070 --> 00:36:51.360 but I sort of agree

913 00:36:51.360 --> 00:36:55.440 because I suspect that you can really do more

914 00:36:55.440 --> 00:36:57.480 with this sort of inner product.

915 00:36:57.480 --> 00:36:59.500 So I'd like to highlight some challenges

916 00:37:00.360 --> 00:37:02.760 associated with this inner product.

917 00:37:02.760 --> 00:37:05.670 So first, let's say, I asked you to design a distribution

918 00:37:05.670 --> 00:37:07.350 with tunable global structure.

919 00:37:07.350 --> 00:37:09.480 So for example, I said, I want you to pick

920 00:37:09.480 --> 00:37:12.060 a generative model or design a distribution of edge flows

921 00:37:12.060 --> 00:37:14.040 that when I sample edge flows from it,

922 00:37:14.040 --> 00:37:18.360 their expected structures matched some expectation.

923 00:37:18.360 --> 00:37:20.910 It's not obvious how to do that given this formula,

924 00:37:21.750 --> 00:37:22.980 it's not obvious in particular

925 00:37:22.980 --> 00:37:24.150 because these projectors,

926 00:37:24.150 --> 00:37:27.090 like the projector on the subspace S typically depend

927 00:37:27.090 --> 00:37:29.910 in fairly non-trivial ways on the graph topology.

928 00:37:29.910 --> 00:37:31.650 So small changes in the graph topology

929 00:37:31.650 --> 00:37:34.350 can completely change as projector.

930 00:37:34.350 --> 00:37:37.350 In essence, it's hard to isolate topology from distribution.

931 00:37:37.350 --> 00:37:38.790 You can think that this inner product,

932 00:37:38.790 --> 00:37:41.313 if I think about it in terms of the IJ entries,

933 00:37:43.110 --> 00:37:46.560 while easy to compute, it's not easy to interpret

934 00:37:46.560 --> 00:37:49.470 because I and J are somewhat arbitrary indexing.

935 00:37:49.470 --> 00:37:51.330 And obviously really the topology of the graph,

936 00:37:51.330 --> 00:37:53.130 it's not encoded in the indexing,

937 00:37:53.130 --> 00:37:56.160 that's encoded in the structure of these matrices.

938 00:37:56.160 --> 00:37:58.680 So in some ways what we really need is a better basis

939 00:37:58.680 --> 00:38:00.330 for computing this inner product.

940 00:38:01.320 --> 00:38:03.090 In addition, computing this inner product

941 00:38:03.090 --> 00:38:05.280 just may not be empirically feasible

942 00:38:05.280 --> 00:38:06.510 because it might not be feasible

943 00:38:06.510 --> 00:38:07.860 to estimate all these covariances.

944 00:38:07.860 --> 00:38:08.760 There's lots of settings

945 00:38:08.760 --> 00:38:10.740 where if you have a random edge flow,

946 00:38:10.740 --> 00:38:12.900 it becomes very expensive to try to estimate

947 00:38:12.900 --> 00:38:14.490 all the covariances in this graph,

948 00:38:14.490 --> 00:38:15.930 err, sorry, in this matrix

949 00:38:15.930 --> 00:38:18.570 because this matrix has as many entries

950 00:38:18.570 --> 00:38:20.793 as there are pairs of edges in the graph.

951 00:38:22.110 --> 00:38:25.650 And typically that number of edges grows fairly quickly

952 00:38:25.650 --> 00:38:27.300 in the number of nodes of the graph.

953 00:38:27.300 --> 00:38:28.770 So in the worst case,

954 00:38:28.770 --> 00:38:30.630 the size of these matrices

955 00:38:30.630 --> 00:38:33.330 goes not to the square of the number of nodes of the graph,

956 00:38:33.330 --> 00:38:34.950 but the number of nodes of the graph to the fourth,

957 00:38:34.950 --> 00:38:37.380 so this becomes very expensive very fast.

958 00:38:37.380 --> 00:38:40.590 Again, we could try to address this problem

959 00:38:40.590 --> 00:38:43.410 if we had a better basis for performing this inner product

960 00:38:43.410 --> 00:38:45.780 because we might hope to be able to truncate

961 00:38:45.780 --> 00:38:47.040 somewhere in that basis,

962 00:38:47.040 --> 00:38:49.190 and use a lower dimensional representation.

963 00:38:50.160 --> 00:38:52.200 So to build there, I'm gonna show you

964 00:38:52.200 --> 00:38:54.930 a particular family of covariances.

965 00:38:54.930 --> 00:38:58.230 We're going to start with a very simple generative model,

966 00:38:58.230 --> 00:39:00.300 so let's suppose that each node of the graph

967 00:39:00.300 --> 00:39:01.860 is assigned some set of attributes,

968 00:39:01.860 --> 00:39:03.523 here a random vector X sampled from a...

969 00:39:03.523 --> 00:39:05.250 So you can think of trait space,

970 00:39:05.250 --> 00:39:07.080 a space of possible attributes,

971 00:39:07.080 --> 00:39:08.970 and these are sampled i.i.d.

972 00:39:08.970 --> 00:39:10.410 In addition, we'll assume

973 00:39:10.410 --> 00:39:12.930 that there exists an alternating function F,

974 00:39:12.930 --> 00:39:17.130 which accepts pairs of attributes and returns a real number.

975 00:39:17.130 --> 00:39:19.230 So this is something that I can evaluate

976 00:39:19.230 --> 00:39:20.910 on the endpoints of an edge,

977 00:39:20.910 --> 00:39:22.683 and return an edge flow value.

978 00:39:24.420 --> 00:39:26.340 In this setting,

979 00:39:26.340 --> 00:39:29.160 everything that I'd shown you before simplifies.

980 00:39:29.160 --> 00:39:32.670 So if my edge flow F is drawn by first sampling

981 00:39:32.670 --> 00:39:33.780 a set of attributes,

982 00:39:33.780 --> 00:39:35.220 and then plugging those attributes

983 00:39:35.220 --> 00:39:39.930 into functions on the edges, then the

984 00:39:39.930 --> 00:39:43.800 mean edge flow is zero, so that F bar goes away,

985 00:39:43.800 --> 00:39:46.080 and the covariance reduces to this form.

986 00:39:46.080 --> 00:39:47.940 So you have a standard form where the covariance

987 00:39:47.940 --> 00:39:51.840 in the edge flow is a function of two scalar quantities,

988 00:39:51.840 --> 00:39:53.010 that's sigma squared in row.

989 00:39:53.010 --> 00:39:56.400 These are both statistics associated with this function

990 00:39:56.400 --> 00:39:59.220 and the distribution of traits.

991 00:39:59.220 --> 00:40:00.180 And then some matrices,

992 00:40:00.180 --> 00:40:01.560 so we have an identity matrix,

993 00:40:01.560 --> 00:40:04.620 and we have this gradient matrix showing up again.

994 00:40:04.620 --> 00:40:07.320 This is really nice because when you plug it back in

995 00:40:07.320 --> 00:40:11.403 to try to compute say the expected sizes of the components,

996 00:40:12.510 --> 00:40:14.880 this matrix inner product

997 00:40:14.880 --> 00:40:16.920 that I was complaining about before,

998 00:40:16.920 --> 00:40:19.290 this whole matrix inner product simplifies.

999 00:40:19.290 --> 00:40:21.060 So when you have a variance

1000 00:40:21.060 --> 00:40:23.400 that's in this nice, simple canonical form,

1001 00:40:23.400 --> 00:40:25.800 then the expected overall size of the edge flow,

1002 00:40:25.800 --> 00:40:27.240 that's just sigma squared,

1003 00:40:27.240 --> 00:40:29.580 the expected size projected onto that

1004 00:40:29.580 --> 00:40:31.030 sort of conservative subspace

1005 00:40:32.250 --> 00:40:34.830 that breaks into this combination

1006 00:40:34.830 --> 00:40:36.840 of the sigma squared in the row.

1007 00:40:36.840 --> 00:40:38.940 Again, those are some simple statistics.

1008 00:40:38.940 --> 00:40:41.430 And then V, E, L, and E,

1009 00:40:41.430 --> 00:40:42.360 those are just sort of

1010 00:40:42.360 --> 00:40:43.453 essentially dimension counting on the network.

1011 00:40:43.453 --> 00:40:46.860 So this is the number of vertices, the number of edges,

1012 00:40:46.860 --> 00:40:47.790 and the number of loops,

1013 00:40:47.790 --> 00:40:49.320 the number of loops that's the number of edges

1014 00:40:49.320 --> 00:40:51.990 minus the number of vertices plus one.

1015 00:40:51.990 --> 00:40:54.720 And similarly, the expected cyclic size

1016 00:40:54.720 --> 00:40:57.240 or size of the cyclic component reduces to,

1017 00:40:57.240 --> 00:40:58.830 again, this sort of scalar factor

1018 00:40:58.830 --> 00:41:00.660 in terms of some simple statistics

1019 00:41:00.660 --> 00:41:03.025 and some dimension counting sort of

1020 00:41:03.025 --> 00:41:05.643 topology related quantities.

1021 00:41:07.375 --> 00:41:10.530 So this is very nice because this allows us

1022 00:41:10.530 --> 00:41:12.900 to really separate the role of topology

1023 00:41:12.900 --> 00:41:14.280 from the role of the generative model.

1024 00:41:14.280 --> 00:41:16.980 The generative model determines sigma in row,

1025 00:41:16.980 --> 00:41:19.323 and topology determines these dimensions.

1026 00:41:21.630 --> 00:41:24.363 It turns out that the same thing is true,

1027 00:41:25.560 --> 00:41:28.590 even if you don't sample the edge flow

1028 00:41:28.590 --> 00:41:31.050 using this sort of trait approach,

1029 00:41:31.050 --> 00:41:32.610 but the graph is complete.

1030 00:41:32.610 --> 00:41:34.380 So if your graph is complete,

1031 00:41:34.380 --> 00:41:36.630 then no matter how you sample your edge flow,

1032 00:41:36.630 --> 00:41:38.280 for any edge flow distribution,

1033 00:41:38.280 --> 00:41:40.350 exactly the same formulas hold,

1034 00:41:40.350 --> 00:41:42.840 you just replace those simple statistics

1035 00:41:42.840 --> 00:41:46.770 with estimators for those statistics given your sample flow.

1036 00:41:46.770 --> 00:41:48.900 And this is sort of a striking result

1037 00:41:48.900 --> 00:41:51.150 because this says that this conclusion

1038 00:41:51.150 --> 00:41:53.730 that was linked to some specific generative model

1039 00:41:53.730 --> 00:41:55.740 with some very sort of specific assumptions, right?

1040 00:41:55.740 --> 00:41:59.100 We assumed it was i.i.d. extends to all complete graphs,

1041 00:41:59.100 --> 00:42:02.193 regardless of the actual distribution that we sampled from.

1042 00:42:04.650 --> 00:42:05.790 Up until this point,

1043 00:42:05.790 --> 00:42:07.790 this is kind of just an algebra miracle.

1044 00:42:09.180 --> 00:42:10.013 And one of the things I'd like to do

1045 00:42:10.013 --> 00:42:12.660 at the end of this talk is explain why this is true,

1046 00:42:12.660 --> 00:42:14.823 and show how to generalize these results.

1047 00:42:16.080 --> 00:42:16.950 So to build there,

1048 00:42:16.950 --> 00:42:19.050 let's emphasize some of the advantages of this.

1049 00:42:19.050 --> 00:42:21.540 So first, the advantages of this model,

1050 00:42:21.540 --> 00:42:23.970 it's mechanistically plausible in certain settings,

1051 00:42:23.970 --> 00:42:27.510 it cleanly separated the role of topology and distribution.

1052 00:42:27.510 --> 00:42:29.880 And these coefficients that had to do with the topology,

1053 00:42:29.880 --> 00:42:30.960 these are just dimensions,

1054 00:42:30.960 --> 00:42:33.510 these are non-negative quantities.

1055 00:42:33.510 --> 00:42:36.030 So it's easy to work out monotonic relationships

1056 00:42:36.030 --> 00:42:37.980 between expected structure

1057 00:42:37.980 --> 00:42:41.073 and simple statistics of the edge flow distribution.

1058 00:42:43.770 --> 00:42:47.010 The fact that you can do that enables more general analysis.

1059 00:42:47.010 --> 00:42:48.240 So what I'm showing you on the right here,

1060 00:42:48.240 --> 00:42:50.730 this is from a different application area.

1061 00:42:50.730 --> 00:42:53.220 This was an experiment where we trained

1062 00:42:53.220 --> 00:42:57.600 a set of agents to play a game using a genetic algorithm,

1063 00:42:57.600 --> 00:43:00.780 and then we looked at the expected sizes of sort of cyclic

1064 00:43:00.780 --> 00:43:04.770 and acyclic components in a tournament among those agents.

1065 00:43:04.770 --> 00:43:07.620 And you could actually predict these curves

1066 00:43:07.620 --> 00:43:09.780 using this sort of type of structural analysis

1067 00:43:09.780 --> 00:43:13.230 because it was possible to predict the dynamics

1068 00:43:13.230 --> 00:43:17.330 of the simple statistics, this sigma in this row.

1069 00:43:17.330 --> 00:43:19.980 So this is a really powerful analytical tool,

1070 00:43:19.980 --> 00:43:22.530 but it is limited to this particular model.

1071 00:43:22.530 --> 00:43:25.590 In particular, it only models unstructured cycles.

1072 00:43:25.590 --> 00:43:26.970 So if you look at the cyclic component

1073 00:43:26.970 --> 00:43:29.940 generated by this model, it just looks like random noise

1074 00:43:29.940 --> 00:43:32.943 that's been projected onto the range of the curl transpose.

1075 00:43:33.870 --> 00:43:36.120 It's limited to correlations on adjacent edges,

1076 00:43:36.120 --> 00:43:38.340 so we only generate correlations on edges

1077 00:43:38.340 --> 00:43:39.420 that share an endpoint

1078 00:43:39.420 --> 00:43:40.950 because you could think that all of the original

1079 00:43:40.950 --> 00:43:43.233 random information comes from the endpoints.

1080 00:43:44.575 --> 00:43:46.560 And then it's in some ways not general enough,

1081 00:43:46.560 --> 00:43:48.060 so it lacks some expressivity.

1082 00:43:48.060 --> 00:43:50.970 We can't parametize all possible expected structures

1083 00:43:50.970 --> 00:43:54.270 by picking a sigma in a row.

1084 00:43:54.270 --> 00:43:55.920 And we lack some notion of sufficiency,

1085 00:43:55.920 --> 00:43:58.410 i.e. if the graph is not complete,

1086 00:43:58.410 --> 00:44:00.840 then this nice algebraic property

1087 00:44:00.840 --> 00:44:02.970 that it actually didn't matter what the distribution was,

1088 00:44:02.970 --> 00:44:04.470 this fails to hold.

1089 00:44:04.470 --> 00:44:06.060 So if the graph is not complete,

1090 00:44:06.060 --> 00:44:09.228 then projection onto the family of covariances

1091 00:44:09.228 --> 00:44:11.430 parameterized in this fashion

1092 00:44:11.430 --> 00:44:13.473 changes the expected global structure.

1093 00:44:14.640 --> 00:44:16.980 So we would like to address these limitations.

1094 00:44:16.980 --> 00:44:18.810 And so our goal for the next part of this talk

1095 00:44:18.810 --> 00:44:21.240 is to really generalize these results.

1096 00:44:21.240 --> 00:44:22.710 To generalize, we're going to

1097 00:44:22.710 --> 00:44:24.930 switch our perspective a little bit.

1098 00:44:24.930 --> 00:44:27.420 So I'll recall this formula

1099 00:44:27.420 --> 00:44:29.730 that if we generate our edge flow

1100 00:44:29.730 --> 00:44:31.650 by sampling quantities on the endpoints,

1101 00:44:31.650 --> 00:44:34.110 and then plugging them into functions on the edges,

1102 00:44:34.110 --> 00:44:35.297 then you necessarily get a covariance

1103 00:44:35.297 --> 00:44:37.320 that's in this two parameter family

1104 00:44:37.320 --> 00:44:38.820 where I have two scalar quantities

1105 00:44:38.820 --> 00:44:40.590 associated with the statistics of the edge flow.

1106 00:44:40.590 --> 00:44:42.210 That's the sigma in this row.

1107 00:44:42.210 --> 00:44:44.160 And then I have some matrices that are associated

1108 00:44:44.160 --> 00:44:45.480 with the topology of the network

1109 00:44:45.480 --> 00:44:47.463 in the subspaces I'm projecting onto.

1110 00:44:48.480 --> 00:44:50.760 These are related to a different way

1111 00:44:50.760 --> 00:44:52.290 of looking at the graph.

1112 00:44:52.290 --> 00:44:54.450 So I can start with my original graph

1113 00:44:54.450 --> 00:44:56.760 and then I can convert it to an edge graph

1114 00:44:56.760 --> 00:44:59.373 where I have one node per edge in the graph,

1115 00:45:00.210 --> 00:45:02.823 and nodes are connected if they share an endpoint.

1116 00:45:04.080 --> 00:45:07.320 You can then assign essentially signs to these edges

1117 00:45:07.320 --> 00:45:10.530 based on whether the edge direction chosen

1118 00:45:10.530 --> 00:45:11.880 in the original graph is consistent

1119 00:45:11.880 --> 00:45:15.810 or inconsistent at the node that links to edges.

1120 00:45:15.810 --> 00:45:19.890 So for example, edges one and two both point into this node,

1121 00:45:19.890 --> 00:45:21.360 so there's an edge that's linking

1122 00:45:21.360 --> 00:45:24.540 one and two in the edge graph with a positive sum.

1123 00:45:24.540 --> 00:45:29.070 This essentially tells you that the influence of

1124 00:45:29.070 --> 00:45:33.240 random information assigned on this node linking one and two

1125 00:45:33.240 --> 00:45:36.210 would positively correlate the sample edge flow

1126 00:45:36.210 --> 00:45:37.323 on edges one and two.

1127 00:45:38.370 --> 00:45:40.770 Then this form, what this form

1128 00:45:40.770 --> 00:45:42.990 sort of for covariance matrices says,

1129 00:45:42.990 --> 00:45:46.200 is that we're looking at families of edge flows

1130 00:45:46.200 --> 00:45:48.690 that have correlations on edges sharing an endpoint.

1131 00:45:48.690 --> 00:45:51.150 So edges at distance one in this edge graph,

1132 00:45:51.150 --> 00:45:52.380 and non-adjacent edges are

1133 00:45:52.380 --> 00:45:54.130 entirely independent of each other.

1134 00:45:56.310 --> 00:45:57.143 Okay?

1135 00:45:58.230 --> 00:45:59.400 So that's essentially what

1136 00:45:59.400 --> 00:46:00.870 the trait performance model is doing, 1137 00:46:00.870 --> 00:46:03.690 is it's parameterizing a family of covariance matrices

1138 00:46:03.690 --> 00:46:05.910 where we're modeling correlations at distance one,

1139 00:46:05.910 --> 00:46:07.590 but not further in the edge graph.

1140 00:46:07.590 --> 00:46:08.820 So then the natural thought

1141 00:46:08.820 --> 00:46:10.800 for how to generalize these results is to ask,

1142 00:46:10.800 --> 00:46:12.840 can we model longer distance correlations

1143 00:46:12.840 --> 00:46:13.790 through this graph?

1144 00:46:15.000 --> 00:46:17.040 To do so, let's think a little bit about

1145 00:46:17.040 --> 00:46:20.970 what this matrix that's showing up inside the covariance is.

1146 00:46:20.970 --> 00:46:23.820 So we have a gradient, tons of gradient transpose.

1147 00:46:23.820 --> 00:46:27.903 This is an effect of Laplacian for that edge graph.

1148 00:46:29.700 --> 00:46:31.680 And you can do this for other motifs.

1149 00:46:31.680 --> 00:46:34.710 If you think about different sort of motif constructions,

1150 00:46:34.710 --> 00:46:38.400 essentially if you take a product of M transpose times M,

1151 00:46:38.400 --> 00:46:40.680 that will generate something that looks like a Laplacian

1152 00:46:40.680 --> 00:46:44.070 or an adjacency matrix for a graph

1153 00:46:44.070 --> 00:46:47.250 where I'm assigning nodes to be motifs

1154 00:46:47.250 --> 00:46:50.190 and looking at the overlap of motifs.

1155 00:46:50.190 --> 00:46:51.990 And if I look at M times M transpose,

1156 00:46:51.990 --> 00:46:54.840 and I'm looking at the overlap of edges via shared motifs.

1157 00:46:54.840 --> 00:46:56.010 So these operators you can think

1158 00:46:56.010 --> 00:46:58.650 about as being Laplacians for some sort of graph

1159 00:46:58.650 --> 00:47:01.413 that's generated from the original graph motifs.

1160 00:47:03.630 --> 00:47:06.480 Like any adjacency matrix,

1161 00:47:06.480 --> 00:47:11.040 powers of something like GG transpose minus 2I,

1162 00:47:11.040 --> 00:47:13.800 that will model connections along longer paths

1163 00:47:13.800 --> 00:47:15.810 along longer distances in these graphs

1164 00:47:15.810 --> 00:47:16.643 associated with motifs,

1165 00:47:16.643 --> 00:47:18.290 in this case with the edge graph.

1166 00:47:19.620 --> 00:47:21.060 So our thought is maybe,

1167 00:47:21.060 --> 00:47:23.280 well, we could extend this trait performance family

1168 00:47:23.280 --> 00:47:26.610 of covariance matrices by instead of only looking at

1169 00:47:26.610 --> 00:47:30.750 a linear combination of an identity matrix, and this matrix,

1170 00:47:30.750 --> 00:47:32.190 we could look at a power series.

1171 00:47:32.190 --> 00:47:36.600 So we could consider combining powers of this matrix.

1172 00:47:36.600 --> 00:47:39.390 And this will generate this family of matrices

1173 00:47:39.390 --> 00:47:41.400 that are parameterized by some set of coefficients-

1174 00:47:41.400 --> 00:47:43.149 <v Robert>Dr. Strang?</v>

1175 00:47:43.149 --> 00:47:44.370 <v ->Ah, yes?</v> <v ->I apologize (mumbles)</v>

1176 00:47:44.370 --> 00:47:45.600 I just wanna remind you

1177 00:47:45.600 --> 00:47:48.240 that we have a rather tight time limit,

1178 00:47:48.240 --> 00:47:50.250 approximately a couple of minutes.

1179 00:47:50.250 --> 00:47:51.303 <v ->Yes, of course.</v>

1180 00:47:52.170 --> 00:47:57.150 So here, the idea is to parametize this family of matrices

1181 00:47:57.150 --> 00:48:00.450 by introducing a set of polynomials with coefficients alpha,

1182 00:48:00.450 --> 00:48:03.420 and then plugging into the polynomial,

1183 00:48:03.420 --> 00:48:06.450 the Laplacian that's generated by sort of the, 1184 00:48:06.450 --> 00:48:07.530 or the adjacent matrix

1185 00:48:07.530 --> 00:48:10.830 generated by the graph motifs we're interested in.

1186 00:48:10.830 --> 00:48:12.030 And that trait performance result,

1187 00:48:12.030 --> 00:48:14.310 that was really just looking at the first order case here,

1188 00:48:14.310 --> 00:48:17.070 that was looking at a linear polynomial

1189 00:48:17.070 --> 00:48:19.680 with these chosen coefficients.

1190 00:48:19.680 --> 00:48:24.120 This power series model is really nice analytically,

1191 00:48:24.120 --> 00:48:28.260 so if we start with some graph operator M,

1192 00:48:28.260 --> 00:48:31.020 and we consider the family of covariance matrices

1193 00:48:31.020 --> 00:48:33.630 generated by plugging M, M transpose

1194 00:48:33.630 --> 00:48:36.240 into some polynomial and power series,

1195 00:48:36.240 --> 00:48:39.240 then this family of matrices is contained

1196 00:48:39.240 --> 00:48:42.213 within the span of powers of M, M transpose.

1197 00:48:45.030 --> 00:48:46.680 You can talk about this family

1198 00:48:46.680 --> 00:48:47.940 sort of in terms of combinatorics.

1199 00:48:47.940 --> 00:48:49.830 So for example, if we use that gradient

1200 00:48:49.830 --> 00:48:52.410 times gradient transpose minus twice the identity,

1201 00:48:52.410 --> 00:48:54.660 then powers of this is essentially, again, paths counting.

1202 00:48:54.660 --> 00:48:56.673 So this is counting paths of length N.

1203 00:48:57.780 --> 00:49:00.270 You can also look at things like the trace of these powers.

1204 00:49:00.270 --> 00:49:01.980 So if you look at the trace series,

1205 00:49:01.980 --> 00:49:03.750 that's the sequence where you look at the trace

1206 00:49:03.750 --> 00:49:06.120 of powers of these,

1207 00:49:06.120 --> 00:49:07.970 essentially these adjacency matrices.

1208 00:49:08.820 --> 00:49:10.770 This is doing some sort of loop count

1209 00:49:10.770 --> 00:49:13.800 where we're counting loops of different length.

1210 00:49:13.800 --> 00:49:14.910 And you could think that this trace series

1211 00:49:14.910 --> 00:49:17.010 in some sense is controlling amplification

1212 00:49:17.010 --> 00:49:20.073 of self-correlations within the sampled edge flow.

1213 00:49:21.840 --> 00:49:22.980 Depending on the generative model,

1214 00:49:22.980 --> 00:49:24.720 we might wanna use different operators

1215 00:49:24.720 --> 00:49:26.040 for generating this family.

1216 00:49:26.040 --> 00:49:27.720 So, for example, going back to that

1217 00:49:27.720 --> 00:49:30.608 synaptic plasticity model with coupled oscillators,

1218 00:49:30.608 --> 00:49:33.570 in this case using the gradient to generate

1219 00:49:33.570 --> 00:49:34.713 the family of covariance matrices.

1220 00:49:34.713 --> 00:49:36.750 It's not really the right structure

1221 00:49:36.750 --> 00:49:39.480 because the dynamics of the model

1222 00:49:39.480 --> 00:49:42.690 sort of have these natural cyclic connections.

1223 00:49:42.690 --> 00:49:45.660 So it's better to build the power series using the curl.

1224 00:49:45.660 --> 00:49:47.130 So depending on your model,

1225 00:49:47.130 --> 00:49:48.840 you can adapt this power series family

1226 00:49:48.840 --> 00:49:50.940 by plugging in a different graph operator.

1227 00:49:52.560 --> 00:49:55.200 Let's see now, what happens if we try to compute

1228 00:49:55.200 --> 00:49:57.810 the expected sizes of some components

1229 00:49:57.810 --> 00:50:00.240 using a power series of this form?

1230 00:50:00.240 --> 00:50:03.570 So if the variance or covariance matrix

1231 00:50:03.570 --> 00:50:05.730 for our edge flow is a power series in,

1232 00:50:05.730 --> 00:50:08.460 for example, the gradient, gradient transpose,

1233 00:50:08.460 --> 00:50:11.580 then the expected sizes of the measures

1234 00:50:11.580 --> 00:50:13.080 can all be expressed as

1235 00:50:13.080 --> 00:50:16.110 linear combinations of this trace series

1236 00:50:16.110 --> 00:50:18.600 and the coefficients of the original polynomial.

1237 00:50:18.600 --> 00:50:21.390 For example, the expected cyclic size of the flow

1238 00:50:21.390 --> 00:50:23.700 is just the polynomial evaluated at negative two

1239 00:50:23.700 --> 00:50:26.130 multiplied by the number of the loops in the graph.

1240 00:50:26.130 --> 00:50:29.040 And this really generalizes that trait performance result

1241 00:50:29.040 --> 00:50:30.900 because the trait performance result is given

1242 00:50:30.900 --> 00:50:33.200 by restricting these polynomials to be linear.

1243 00:50:34.050 --> 00:50:34.883 Okay?

1244 00:50:36.270 --> 00:50:39.693 This you can extend sort of to other bases,

1245 00:50:41.310 --> 00:50:43.410 but really what this accomplishes is

1246 00:50:43.410 --> 00:50:45.210 by generalizing trait performance,

1247 00:50:45.210 --> 00:50:50.210 we achieve this sort of generic properties

1248 00:50:50.400 --> 00:50:52.140 that it failed to have.

1249 00:50:52.140 --> 00:50:55.560 So in particular, if I have an edge flow subspace S

1250 00:50:55.560 --> 00:50:58.740 spanned by a set of flow motifs stored in some operator M,

1251 00:50:58.740 --> 00:51:00.590 then this power series family of covariance

1252 00:51:00.590 --> 00:51:03.300 is associated with the Laplacian,

1253 00:51:03.300 --> 00:51:07.440 that is M times M transpose is both expressive

1254 00:51:07.440 --> 00:51:10.950 in the sense that for any non-negative A and B,

1255 00:51:10.950 --> 00:51:13.380 I can pick some alpha and beta,

1256 00:51:13.380 --> 00:51:16.020 so that the expected size of the projection of F

1257 00:51:16.020 --> 00:51:17.700 onto the subspace is A,

1258 00:51:17.700 --> 00:51:19.440 and the projected size of F

1259 00:51:19.440 --> 00:51:22.390 onto the subspace orthogonal to S is B

1260 00:51:23.340 --> 00:51:26.133 for any covariance in this power series family.

1261 00:51:27.060 --> 00:51:29.160 And it's sufficient in the sense

1262 00:51:29.160 --> 00:51:31.170 that for any edge flow distribution

1263 00:51:31.170 --> 00:51:34.710 with mean zero in covariance V.

1264 00:51:34.710 --> 00:51:37.980 If C is the matrix nearest to V in Frobenius norm

1265 00:51:37.980 --> 00:51:40.380 restricted to the power series family,

1266 00:51:40.380 --> 00:51:43.770 then these inner products computed in terms of C

1267 00:51:43.770 --> 00:51:45.570 are exactly the same as the inner products

1268 00:51:45.570 --> 00:51:47.070 computed in terms of V,

1269 00:51:47.070 --> 00:51:49.020 so they directly predict the structure,

1270 00:51:49.020 --> 00:51:51.390 which means that if I use this power series family,

1271 00:51:51.390 --> 00:51:53.580 discrepancies off of this family

1272 00:51:53.580 --> 00:51:55.380 don't change the expected structure.

1273 00:51:56.520 --> 00:51:57.353 Okay?

1274 00:51:57.353 --> 00:51:59.010 So I know I'm short on time here,

1275 00:51:59.010 --> 00:52:02.790 so I'd like to skip then just to the end of this talk.

1276 00:52:02.790 --> 00:52:04.200 There's further things you can do with this,

1277 00:52:04.200 --> 00:52:05.610 this is sort of really nice.

1278 00:52:05.610 --> 00:52:08.460 Mathematically you can build an approximation theory

1279 00:52:08.460 --> 00:52:11.730 out of this and study for different random graph families,

1280 00:52:11.730 --> 00:52:14.820 how many terms in these power series you need?

1281 00:52:14.820 --> 00:52:16.800 And those terms define some nice,

1282 00:52:16.800 --> 00:52:18.570 sort of simple minimal set of statistics

1283 00:52:18.570 --> 00:52:20.433 to try to sort of estimate structure,

1284 00:52:22.110 --> 00:52:24.490 but I'd like to really just get to the end here

1285 00:52:25.350 --> 00:52:28.260 and emphasize the takeaways from this talk.

1286 00:52:28.260 --> 00:52:29.580 So the first half of this talk

1287 00:52:29.580 --> 00:52:32.130 was focused on information flow.

1288 00:52:32.130 --> 00:52:35.160 What we saw is that information flow is a non-trivial,

1289 00:52:35.160 --> 00:52:36.810 but well studied, estimation problem.

1290 00:52:36.810 --> 00:52:38.310 And this is something that at least on my side

1291 00:52:38.310 --> 00:52:40.530 sort of is a work in progress with students.

1292 00:52:40.530 --> 00:52:43.380 Here in some ways, the conclusion of that first half

1293 00:52:43.380 --> 00:52:44.820 would be that causation entropy

1294 00:52:44.820 --> 00:52:46.890 may be a more appropriate measure than TE

1295 00:52:46.890 --> 00:52:48.540 when trying to build these flow graphs

1296 00:52:48.540 --> 00:52:51.240 to apply these structural measures to.

1297 00:52:51.240 --> 00:52:53.160 Then on the structural side,

1298 00:52:53.160 --> 00:52:54.540 we can say that power series family,

1299 00:52:54.540 --> 00:52:56.610 this is a nice family of covariance matrices.

1300 00:52:56.610 --> 00:52:59.490 It has nice properties that are useful empirically

1301 00:52:59.490 --> 00:53:01.830 because they let us build global correlation structures

1302 00:53:01.830 --> 00:53:03.450 from a sequence of local correlations

1303 00:53:03.450 --> 00:53:04.683 from that power series.

1304 00:53:06.240 --> 00:53:08.220 If you plug this back into the expected measures,

1305 00:53:08.220 --> 00:53:09.990 you can recover monotonic relations,

1306 00:53:09.990 --> 00:53:12.180 like in that limited trait performance case.

1307 00:53:12.180 --> 00:53:14.400 And truncation of these power series

1308 00:53:14.400 --> 00:53:15.840 reduces the number of quantities

1309 00:53:15.840 --> 00:53:17.663 that you would actually need to measure.

1310 00:53:18.600 --> 00:53:19.890 Actually to a number of quantities

1311 00:53:19.890 --> 00:53:22.080 that can be quite small relative to the graph, 1312 00:53:22.080 --> 00:53:24.353 and that's where this approximation theory comes in.

1313 00:53:25.290 --> 00:53:28.140 One way, sort of maybe to summarize this entire approach

1314 00:53:28.140 --> 00:53:30.810 is what we've done is by looking at these power series

1315 00:53:30.810 --> 00:53:33.030 built in terms of the graph operators

1316 00:53:33.030 --> 00:53:35.460 is it provides a way to study

1317 00:53:35.460 --> 00:53:38.100 inherently heterogeneous connections,

1318 00:53:38.100 --> 00:53:40.530 or covariances, or edge flow distributions

1319 00:53:40.530 --> 00:53:42.630 using a homogeneous correlation model

1320 00:53:42.630 --> 00:53:44.670 that's built sort of at multiple scales

1321 00:53:44.670 --> 00:53:47.553 by starting the local scale, and then looking at powers.

1322 00:53:48.960 --> 00:53:49.953 In some ways this is a comment

1323 00:53:49.953 --> 00:53:53.310 that I ended a previous version of this talk with.

1324 00:53:53.310 --> 00:53:55.590 I still think that this structural analysis is in some ways

1325 00:53:55.590 --> 00:53:57.270 a hammer seeking a nail,

1326 00:53:57.270 --> 00:53:59.160 and that this inflammation flow construction,

1327 00:53:59.160 --> 00:54:02.100 this is work in progress to try to build that nail.

1328 00:54:02.100 --> 00:54:04.110 So thank you all for your attention,

1329 00:54:04.110 --> 00:54:05.913 I'll turn it now over to questions.

1330 00:54:08.892 --> 00:54:12.573 <v Robert>(mumbles) really appreciate it.</v>

1331 00:54:14.130 --> 00:54:15.600 Unfortunately, for those of you on Zoom,

1332 00:54:15.600 --> 00:54:17.280 you're welcome to keep up the conversation,

1333 00:54:17.280 --> 00:54:19.890 so (mumbles) unfortunately have to clear the room.

1334 00:54:19.890 --> 00:54:23.100 So I do apologize (mumbles)

1335 00:54:24.685 --> 00:54:25.768 Dr. Steinman?

1336 00:54:26.643 --> 00:54:28.359 It might be interesting, yeah. (laughs)

1337 00:54:28.359 --> 00:54:30.330 (students laugh)

1338 00:54:30.330 --> 00:54:33.330 Dr. Strang? <v ->Oh, yes, yeah.</v>

1339 00:54:33.330 --> 00:54:34.710 <v Robert>Okay, do you mind if people...?</v>

1340 00:54:34.710 --> 00:54:35.717 Yeah, we have to clear the room,

1341 00:54:35.717 --> 00:54:39.613 do you mind if people email you if they have questions?

1342 00:54:39.613 --> 00:54:42.060 <v ->I'm sorry, I couldn't hear the end of the question.</v>

1343 00:54:42.060 --> 00:54:43.213 Do I mind if...?

1344 00:54:45.060 --> 00:54:46.530 <v Robert>We have to clear the room,</v>

1345 00:54:46.530 --> 00:54:49.027 do you mind if people email you if they have questions?

1346 00:54:49.027 --> 00:54:49.884 <v ->No, not at all.</v>

1347 00:54:49.884 --> 00:54:52.110 <v Robert>(mumbles) may continue the conversation,</v>

1348 00:54:52.110 --> 00:54:54.330 so I do apologize, they are literally

1349 00:54:54.330 --> 00:54:56.760 just stepping in the room right now.

1350 00:54:56.760 --> 00:54:58.644 <v ->Okay, no, yeah, that's totally fine.</v>

1351 00:54:58.644 --> 00:55:00.660 <v Robert>Thank you, thank you.</v>

1352 00:55:00.660 --> 00:55:02.820 And thanks again for a wonderful talk.

1353 00:55:02.820 --> 00:55:03.653 <v ->Thank you.</v>