Capacity of Continuous Channels with Memory via Directed Information Neural Estimator

Ziv Aharoni
Ben Gurion University
zivah@post.bgu.ac.il

Dor Tsur
Ben Gurion University
dortz@post.bgu.ac.il

Ziv Goldfeld
Cornell University
goldfeld@cornell.edu

Haim H. Permuter
Ben Gurion University
haimp@bgu.ac.il

Abstract—Calculating the capacity (with or without feedback) of channels with memory and continuous alphabets is a challenging task. It requires optimizing the directed information rate over all channel input distributions. The objective is a multi-letter expression, whose analytic solution is only known for a few specific cases. When no analytic solution is present or the channel model is unknown, there is no unified framework for calculating or even approximating capacity. This work proposes a novel capacity estimation algorithm that treats the channel as a ‘black-box’, both when feedback is or is not present. The algorithm has two main ingredients: (i) a neural distribution transformer (NDT) model that shapes a noise variable into the channel input distribution, which we are able to sample, and (ii) the directed information neural estimator (DINE) that estimates the communication rate of the current NDT model. These models are trained by an alternating maximization procedure to both estimate the channel capacity and obtain an NDT for the optimal input distribution. The method is demonstrated on the moving average additive Gaussian noise channel, where it is shown that both the capacity and feedback capacity are estimated without knowledge of the channel transition kernel. The proposed estimation framework opens the door to a myriad of capacity approximation results for continuous alphabet channels that were inaccessible until now.

I. INTRODUCTION

Many discrete-time continuous-alphabet communication channels involve correlated noise or inter-symbol interference (ISI). Two predominant communication scenarios over such channels are when feedback from the receiver back to the transmitter is or is not present. The fundamental rates of reliable communication over such channels are, respectively, the feedback (FB) and feedforward (FF) capacity. Starting from the latter, the FF capacity of an \( n \)-fold point-to-point channel \( P_{Y^n|X^n} \), denoted \( C_{FF} \), is given by [11]

\[
C_{FF} = \lim_{n \to \infty} \sup_{P_{X^n}} \frac{1}{n} I(X^n; Y^n). \tag{1}
\]

In the presence of feedback, the FB capacity \( C_{FB} \) is [17]

\[
C_{FB} = \lim_{n \to \infty} \sup_{P_{X^n,Y^n-1}} \frac{1}{n} I(X^n \to Y^n) \tag{2}
\]

where,

\[
I(X^n \to Y^n) := \sum_{i=1}^{n} I(X^i; Y^i|Y^{i-1}) \tag{3}
\]

is the directed information (DI) from the input sequence \( X^n \) to the output \( Y^n \) [8], and \( P_{X^n,Y^n-1} := \prod_{i=1}^{n} P_{X^i,X^{i-1}|Y^{i-1}} \) is the distribution of \( X^n \) causally-conditioned on \( Y^{n-1} \) (see [21], [24] for further details). Built on [3], for stationary processes, the DI rate is defined as

\[
I(X \to Y) := \lim_{n \to \infty} \frac{1}{n} I(X^n \to Y^n). \tag{4}
\]

As proved in [8], when feedback is not present, the optimization problem (2) performed over the marginals \( P_{X^n} \) is equivalent to the optimization in (1). This casts DI as a unifying information measure for representing both FF and FB capacities.

Computing \( C_{FF} \) and \( C_{FB} \) requires solving a multi-letter optimization problem. Closed form solutions to this challenging task are known only in several special cases. A common example for \( C_{FF} \) is the Gaussian channel with memory [14] and the ISI Gaussian channel [13]. There are no known extensions of these solutions to the non-Gaussian case. For \( C_{FB} \), a solution for the 1st order moving average additive Gaussian noise (MA(1)-AGN) channel was found [12]. Another closed form characterization is available for auto-regressive moving-average (ARMA) AGN channels [11]. To the best of our knowledge, these are the only two non-trivial examples of continuous channels with memory whose FB capacity is known in closed form. Furthermore, when the channel model is unknown, there is no efficient method for numerically approximating capacity.

Some recent progress related to capacity computation was made based on deep learning (DL) techniques [9], [19]. In a novel work [9], mutual information neural estimator (MINE) [2] was used to learn a modulation for a memoryless channel. In [19], a capacity estimator was proposed based on reinforcement learning algorithm that iteratively estimates and maximizes the DI rate, but only for discrete alphabet channels with a known channel model.

Inspired by the above, we develop the framework for estimating FF and FB capacity of arbitrary continuous-alphabet channels, possible with memory, without knowing the channel model. Our method does not need to know the channel transition kernel. We only assume a stationary channel model and that channel outputs can be sampled by feeding it with inputs. Central to our method are a new DI neural estimator (DINE), used to evaluate the communication rate, and a neural distribution transformer (NDT), used to simulate input distributions. Together, the DINE and NDT
lay the groundwork for our capacity estimation algorithm. In the remainder of this section, we describe DINE, NDT, and their integration into the capacity estimator.

A. Directed Information Neural Estimation

The estimation of mutual information (MI) from samples using neural networks (NNs) is a recently proposed approach \([2, 3]\). It is especially effective when the involved random variables (RVs) are continuous. The concept originated from \([2]\), where MINE was proposed. The core idea is to represent MI using the Donsker-Varadhan (DV) variational formula

\[
I(X; Y) = \sup_{T: X \rightarrow Y \rightarrow R} \mathbb{E}[T(X, Y)] - \log \mathbb{E} \left[e^{T(\tilde{X}, \tilde{Y})}\right],
\]

where \((X, Y) \sim P_{XY}\) and \((\tilde{X}, \tilde{Y}) \sim P_X \otimes P_Y\). The supremum is over all measurable functions \(T\) for which both expectations are finite. Parameterizing \(T\) by an NN and replacing expectations with empirical averages, enables gradient ascent optimization to estimate \(I(X; Y)\). A variant of MINE that goes through estimating the underlying entropy terms was proposed in \([3]\). The new estimators were shown empirically to perform extremely well, especially for continuous alphabets.

Herein, we propose a new estimator for the DI rate \(I(\mathcal{X} \rightarrow \mathcal{Y})\). The DI is factorized as

\[
I(X^n \rightarrow Y^n) = h(Y^n) - h(Y^n \| X^n),
\]

where \(h(Y^n)\) is the differential entropy of \(Y^n\) and \(h(Y^n \| X^n) := \sum_{i=1}^n h(Y_i | Y^{i-1}, X_i)\). Applying the approach of \([3]\) to the entropy terms, we expand each as a Kullback-Leibler (KL) divergence and a cross-entropy (CE) residual and invoke the DV representation. To account for memory, we derive a formula valid for causally dependent data, which involves RNNs as function approximator (rather than the FF network used in the independently and identically distributed (i.i.d.) case). Thus, the DINE is an RNN-based estimator for the directed information rate from \(X^n\) to \(Y^n\) based on their samples.

DI estimators were recently presented in \([25-27]\). Also, an estimator of the transfer entropy using FF networks was proposed \([16]\), which upper bounds the DI in the special case of a jointly Markov process with finite memory. DINE is the first method based on RNN and hence does not assume any parametric model such as discrete alphabets, or Markovity. Further details on the DINE algorithm are given in subsection II-A.

B. Neural Distribution Transformer and Capacity Estimation

DINE accounts for one of the two tasks involved in estimating capacity, it estimates the objective of \([2]\). The remaining task is to optimize this objective over input distributions. Generally, sampling from an arbitrary distribution is a complex task. To overcome this, we design a deep generative model of the channel input distributions, namely the NDT. The idea is similar to ones used for generative modeling tasks, e.g. generative adversarial networks \([23]\) or variational autoencoders \([22]\). The designed NDT maps i.i.d. noise into samples of the channel input distribution. For estimating FB capacity, in addition to the i.i.d. noise, the NDT also receives channel FB as inputs. Together, NDT and DINE form the overall system that estimates the capacity as shown in Fig 1.

The capacity estimation algorithm trains the DINE and NDT models together via an alternating maximization procedure. Namely, we iteratively train each model while keeping the (parameters of the) other one fixed. DINE estimates the communication rate of a fixed NDT input distribution, and the NDT is trained to increase its rate with respect to fixed DINE model. Proceeding until convergence, this results in the capacity estimate, as well as an NDT generative model for the achieving input distribution.

We demonstrate our method on the MA(1)-AGN channel. Both \(C_{\text{FF}}\) and \(C_{\text{FB}}\) are estimated using the same algorithm, using the channel as a black-box to solely generate samples. The estimation results are compared with the analytic solution to show the effectiveness of the proposed approach.

![Fig. 1. The overall capacity estimation system. NDT generates samples that are fed into the channel. DINE uses these samples to improve its estimation of the communication rate. DINE then supplies gradient for the optimization of NDT.](https://github.com/zivaharoni/capacity-estimator-via-dine)
where $h_{CE}(P, Q)$ and $D_{KL}(P \parallel Q)$ are, respectively, the cross entropy (CE) and KL divergence between $P$ and $Q$, and $P_Y$ is uniform reference measure over the support of the dataset. To simplify notation, we use the shorthands

$$D_Y^{(n)} := D_{KL}(P_{Y^n} \parallel P_{Y^{n-1} \otimes P_Y})$$

$$D_Y^{(n)}_{|X} := D_{KL}(P_{Y^n|X^n} \parallel P_{Y^{n-1}|X^{n-1} \otimes P_Y}). \quad (8)$$

Subtracting both elements in $(8)$ and observing that the difference of CE terms equals the DI at the former time step, we have

$$I(X^n \rightarrow Y^n) = I(X^{n-1} \rightarrow Y^{n-1}) + D_Y^{(n)} - D_Y^{(n)}_{|X}. \quad (9)$$

Note that the difference of KL divergences equals $I(X^n; Y^n|Y^{n-1})$. For stationary data processes we take the limit and obtain

$$\lim_{n \rightarrow \infty} D_Y^{(n)} = \lim_{n \rightarrow \infty} I(X^n; Y^n|Y^{n-1}) = I(X \rightarrow Y). \quad (10)$$

Each $D_{KL}$ is expanded by its DV representation [4] as:

$$D_{Y}^{(n)} = \sup_{T: \Omega \rightarrow \mathbb{R}} \mathbb{E}[T(Y^n)] - \log \mathbb{E}[e^{T(Y^n-1, Y)}]$$

$$D_{Y|X}^{(n)} = \sup_{T: \Omega \rightarrow \mathbb{R}} \mathbb{E}[T(Y^n|X^n)] - \log \mathbb{E}[e^{T(Y^n-1|X^{n-1}, Y)}]. \quad (11)$$

To maximize $(11)$, each DV potential is parametrized by a modified LSTM and expected values are estimated by empirical averages over the dataset $D_n := \{(x_i, y_i)\}_{i=1}^n$. Thus, the optimization objectives are:

$$\tilde{D}_{Y|X}(\theta_{Y|X}, D_n) := \frac{1}{n} \sum_{i=1}^{n} T_{\theta_{Y|X}}(y_i|x_i^{(n)}|y_{i-1}^{(n)} - \log \left( \frac{1}{n} \sum_{i=1}^{n} e^{T_{\theta_{Y|X}}(y_{i}|x_{i}^{(n)}|y_{i-1}^{(n)})} \right)$$

$$\tilde{D}_{Y}(\theta_{Y}, D_n) := \frac{1}{n} \sum_{i=1}^{n} T_{\theta_{Y}}(y_{i}|y_{i-1}^{(n)} - \log \left( \frac{1}{n} \sum_{i=1}^{n} e^{T_{\theta_{Y}}(y_{i}|y_{i-1}^{(n)})} \right) \quad (12)$$

where $\tilde{y}_{i} \overset{i.i.d.}{\sim} P_{\tilde{Y}}$ and $T_{\theta_{Y}}, T_{\theta_{Y|X}}$ are the parametrized potentials.

The estimator is given by:

$$\hat{I}_{D_n}(X \rightarrow Y) := \sup_{\theta_{Y|X} \in \theta_{Y|X}} \tilde{D}_{Y|X} - \sup_{\theta_{Y} \in \theta_{Y}} \tilde{D}_{Y}. \quad (13)$$

By universal approximation of RNNs [6] and Breiman’s theorem [7], the maximizer of $(13)$ approaches $I(X \rightarrow Y)$ as the number of samples grows, provided the neural networks are sufficiently expressive.

To capture the time dependencies in $D_n$, we need to introduce a modified LSTM network model for functional approximation. LSTM [5] is an RNN that receives a time series $\{y_{i}\}_{i=1}^{T}$ as input and for each $i$, performs a recursive non-linear transform to calculate its hidden state $s_i$. We denote the LSTM function by $F : (y_{i}, s_{i-1}) \mapsto s_{i}$. The full characterization of $F$ is provided in [5].

We modify the structure of the LSTM to perform the calculations:

$$s_{i} = F(y_{i}, s_{i-1}) = s(y_{i}|y_{i-1}^{(n)})$$

$$\tilde{s}_{i} = F(\tilde{y}_{i}, \tilde{s}_{i-1}) = s(y_{i}|y_{i-1}^{(n)}) \quad (14)$$

A similar modification is introduced for $\tilde{D}_{Y|X}$ by substitution of $y_{i}$ with $(y_{i}, x_{i})$ and $\tilde{y}_{i}$ with $(\tilde{y}_{i}, x_{i})$, we have:

$$s_{i} = F(y_{i}, x_{i}, s_{i-1}) = s(y_{i}|y_{i-1}^{(n)}, x^{i})$$

$$\tilde{s}_{i} = F(\tilde{y}_{i}, x_{i}, \tilde{s}_{i-1}) = s(y_{i}|y_{i-1}^{(n)}, x^{i}) \quad (15)$$

A visualization of a modified LSTM cell (unrolled) is shown in Fig. 2. The LSTM cell’s output is the sequence $\{(s_{i}, \tilde{s}_{i})\}_{i=1}^{n}$, which is fed into a fully-connected layer to obtain $T_{\theta_{Y}}$ and $T_{\theta_{Y|X}}$. As demonstrated by Algorithm 1 and Fig. 2 in each iteration we draw $D_{B}$, a subset on $D_{n}$, of size B. We feed the NN with $D_{B}$ to acquire $T_{\theta_{Y}}, T_{\theta_{Y|X}}$. Those enter the NN loss function $(12)$, and gradients are calculated to update the NN parameters $\theta_{Y}, \theta_{Y|X}$.

**Algorithm 1 Directed Information Rate Estimation**

**input:** Samples of the process $D_{n}$.

**output:** $\hat{I}_{D_{n}}(X \rightarrow Y)$, estimated directed information rate.

**Step 1,** Optimization:

repeat

Draw a batch $D_{B} = \{(x_{i}^{(n)}T, y_{i}^{(n)}T)\}_{i=1}^{B}$

Feed the network with the examples and compute loss $\tilde{D}_{Y|X}(\theta_{Y|X}, D_{B})$.

Update networks parameters:

$$\theta_{Y|X} \leftarrow \theta_{Y|X} + \nabla \tilde{D}_{Y|X}(\theta_{Y|X}, D_{B})$$

$$\theta_{Y} \leftarrow \theta_{Y} + \nabla \tilde{D}_{Y}(\theta_{Y}, D_{B})$$

until convergence

**Step 2,** Perfrom a Monte Carlo estimation over $D_{n}$ and subtract loss evaluations to obtain estimation :

$$\hat{I}_{D_{n}}(X \rightarrow Y) = \tilde{D}_{Y|X}(\theta_{Y|X}, D_{n}) - \tilde{D}_{Y}(\theta_{Y}, D_{n})$$

![Fig. 2. The modified LSTM cell unrolled in the DINE architecture of $\tilde{D}_{Y}$. Recursively, at each time $i$, $(y_{i}, s_{i-1})$ and $(\tilde{y}_{i}, \tilde{s}_{i-1})$ are mapped to $s_{i}$ and $\tilde{s}_{i}$, respectively.](image-url)
Fig. 3. End-to-end architecture for the estimation of \( D_Y(\theta_Y, \mathcal{D}_D) \). Each batch of time sequences enters the system, a batch of the same size is sampled from the reference measure and those enter the NN to compute \( T_{\theta_Y} \) and \( T_{\theta_Y}^{\mathcal{D}}(\theta_Y, \mathcal{D}_D) \).

### B. Neural Distribution Transformer

The DINE model is an effective approach to estimate the argument of \( \mathcal{R} \). However, finding the capacity comprises maximization of the DI with respect to the input distribution. For this purpose we present the NDT model that represents maximization of the DI with respect to the input distribution. The NDT model is fed with i.i.d. noise and its output is the samples of this algorithm on continuous alphabet channels.

Fig. 4. The NDT. The noise and past channel output (if feedback is applied) are fed into an NN. The last layer performs normalization to obey the power constraint, if needed.

### C. Complete Architecture Layout

Combining DINE and NDT models into a complete system enables capacity estimation. As shown in Fig. 1 the NDT model is fed with i.i.d. noise and its output is the samples \( X^n \). These samples are fed into the channel to generate its output. Then, \((X^n, Y^n)\) are fed both to the DINE model that outputs \( \hat{D}_Y(\mathcal{X} \rightarrow \mathcal{Y}) \). To estimate the capacity, DINE and NDT models are trained together. The training scheme, as shown in Algorithm 2, is a variant of alternated maximization procedure. This procedure iterates between updating the DINE and NDT models parameters sets \( \theta, \mu \), where each iteration the parameters of one model are fixed and the other ones are updated. The end of training a long Monte-Carlo evaluation of \( N \) is done in order to estimate the expectations in (12) accurately.

Applying this algorithm to channels with memory estimates their capacity without any specific knowledge of the channel underlying distribution. Next, we demonstrate the effectiveness of this algorithm on continuous alphabet channels.

### III. Numerical Results

We demonstrate the performance of Algorithm 2 on the AWGN channel and the first order MA-AGN channel. The numerical results are then compared with the analytic solution to verify the effectiveness of our method.

#### A. AWGN channel

The power constrained AWGN channel is investigated as an instance of memoryless continuous alphabet channel for which analytic solution is known. The channel model is given by

\[
Y_i = X_i + Z_i, \quad i \in \mathbb{N},
\]

where \( Z_i \sim \mathcal{N}(0, \sigma^2) \) are i.i.d RVs, and \( X_i \) is the channel input sequence bound to the power constraint \( \mathbb{E}[X_i^2] \leq P \). Its capacity is given by \( C = \frac{1}{2} \log(1 + \frac{P}{\sigma^2}) \). In our implementation we chose \( \sigma^2 = 1 \) and estimated the capacity for a range of \( P \) values. The numerical results are compared to the analytic solution in Fig. 5.

#### B. Gaussian MA(1) channel

The calculation of capacity of linear Gaussian channels with memory can be divided into two cases, feedback \( C_{FB} \) and feed-forward \( C_{FB} \) capacity. We will focus on the MA(1) Gaussian channel model, which is given by:

\[
Z_i = \alpha U_{i-1} + U_i \\
Y_i = X_i + Z_i
\]

#### Algorithm 2 Capacity Estimation

**input:** Continuous channel, feedback indicator

**output:** \( \hat{I}_{\mathcal{D}_D}(\mathcal{X} \rightarrow \mathcal{Y}, \mu) \), estimated capacity.

- Initialize DINE parameters, \( \theta_Y, \theta_Y^{\mathcal{X}} \)
- Initialize NDT parameters \( \mu \)
- if feedback indicator
- Add feedback to NDT
- repeat
  - **Step 1: Train DINE model**
    - Generate \( B \) sequences of length \( T \) of i.i.d random noise
    - Compute \( \mathcal{D}_B = \{ (x_i^T, u_i^T) \} \) with NDT and channel
    - Compute \( \hat{D}_Y^{\mathcal{X}}(\theta_Y^{\mathcal{X}}, \mathcal{D}_B), \hat{D}_Y(\theta_Y, \mathcal{D}_B) \)
    - Update DINE parameters:
      \[
      \theta_Y^{\mathcal{X}} \leftarrow \theta_Y^{\mathcal{X}} + \nabla \hat{D}_Y^{\mathcal{X}}(\theta_Y^{\mathcal{X}}, \mathcal{D}_B)
      \]
      \[
      \theta_Y \leftarrow \theta_Y + \nabla \hat{D}_Y(\theta_Y, \mathcal{D}_B)
      \]
  - **Step 2: Train NDT**
    - Generate \( B \) sequences of length \( T \) of i.i.d random noise
    - Compute \( \mathcal{D}_B = \{ (x_i^T, u_i^T) \} \) with NDT and channel
    - Compute the objective:
      \[
      \hat{I}_{\mathcal{D}_D}(\mathcal{X} \rightarrow \mathcal{Y}, \mu) = \hat{D}_Y^{\mathcal{X}}(\theta_Y^{\mathcal{X}}, \mathcal{D}_B) - \hat{D}_Y(\theta_Y, \mathcal{D}_B)
      \]
    - Update NDT parameters:
      \[
      \mu \leftarrow \mu + \nabla_{\mu} \hat{I}_{\mathcal{D}_D}(\mathcal{X} \rightarrow \mathcal{Y}, \mu)
      \]
    - until convergence
- Monte Carlo evaluation of \( \hat{I}_{\mathcal{D}_D}(\mathcal{X} \rightarrow \mathcal{Y}, \mu) \)

return \( \hat{I}_{\mathcal{D}_D}(\mathcal{X} \rightarrow \mathcal{Y}, \mu) \)
where, \( U_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 1) \), \( X_i \) is the channel input sequence bound to the power constraint \( \mathbb{E}[X_i^2] \leq P \), and \( Y_i \) is the channel output.

1) Feed-forward capacity: For the LTI Gaussian channel with input power constraint, \( C_{\text{FF}} \) can be obtained by applying the water-filling algorithm [14]. We applied Algorithm 2 to estimate \( C_{\text{FF}} \) and compare with results of the water-filling algorithm. Results are in Fig. 6.

2) Feedback capacity: In general, \( C_{\text{FB}} \) of the ARMA(k) Gaussian channel can be formulated as a dynamic programming problem, which can be solved by an iterative algorithm [11]. For the particular case of (17), \( C_{\text{FB}} \) is given by \(- \log(x_0)\), where \( x_0 \) is a solution of a 4th order polynomial equation. We applied Algorithm 2 for the feedback capacity to obtain an estimate of \( C_{\text{FB}} \). The results and compared with the analytic solution as shown in Fig. 7.

IV. Conclusion and Future Work

We have presented a methodology to estimate FF and FB capacity using the channel as a "black-box". The estimator is designed by a novel DI estimator (DINE) and NDT model, both based on RNNs. The performance of the estimator are demonstrated on the AWGN and MA(1)-AGN channels, and estimation agrees with the analytic solution.

We wish to further generalize our method of information rate estimation for multi-user communication channels, a field with many unsolved problems and to find theoretical guarantees of the estimator. In addition, information theory (e.g. channel capacity) give us a rigorous mathematical framework where analytical solution are known due to Shannon theory hence this can be a good problem for evaluating machine learning approaches.

REFERENCES

[1] R. G. Gallager. Information theory and reliable communication. Vol. 2. New York: Wiley, 1968.
[2] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm. Mine: mutual information neural estimation. arXiv preprint arXiv:1801.04062. June 2018.

[3] C. Chan, A. Al-Bashabesh, H. P. Huang, M. Lim, D. S. H. Tam and C. Zhao. Neural Entropic Estimation: A faster path to mutual information estimation. arXiv preprint arXiv:1905.12957. May 2019.

[4] M. Donsker, and S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time, iv. Communications on Pure and Applied Mathematics, 36(2):183-212. March 1983.

[5] S. Hochreiter and J. Schmidhuber.Long short-term memory. Neural Computation 9(8): 1735-1780. November 1997.

[6] A. M. Schifer and H. G. Zimmermann. Recurrent neural networks are universal approximators. International journal of neural systems 17.04: 253-263. 2007.

[7] L. Breiman. “The individual ergodic theorem of information theory” The Annals of Mathematical Statistics: 809-811. September 1957. Information Theory, IEEE Trans. Comm., vol. COM-21, pp. 1345-1351. December 1973.

[8] J. Massey. Causality, feedback, and directed information. Proc. Int. Symp. Inf. Theory Appl., pp. 303305. November 1990.

[9] R. Fritschek, R. F. Schaefker, and G. Wunder. Deep Learning for Channel Coding via Neural Mutual Information Estimation. arXiv preprint arXiv:1903.02865. March 2019.

[10] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. Neural networks 2.5 : 359-366. March 1989.

[11] S. Yang, A. Kavcic, and S. Tatikonda. On the feedback capacity of power-constrained Gaussian noise channels with memory. IEEE Trans. Inf. Theory 53.3 : 929-954. March 2007.

[12] C. J. Quinn, T. P. Coleman, N. Kiyavash et al. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. J Comput Neurosci 30, 1744. June 2010.