Transformers for Information Measures Estimation in Time Series Domain

RESEARCH PROPOSAL

OMER LUXEMBOURG · ADVISOR: PROF. HAIM PERMUTER · AUGUST 2025

Why Transfer Entropy (TE)?

TE measures directed information flow between two jointly stationary processes, for a fixed memory length k, l:

$$\mathsf{TE}_{X \to Y}(k, l) := \mathsf{I}\left(X^{l}; Y_{l} | Y_{l-k}^{l-1}\right).$$

Captures causal influence beyond correlation or Mutual Information (MI)

Useful in neuroscience, communications, IoT, finance

During the presentation $l = k$, unless stated otherwise.

Challenges in TE Estimation

Long contexts, continuous signals, non-Gaussian noise

Classical methods (KDE/kNN/copula) suffer from bias/variance trade-off

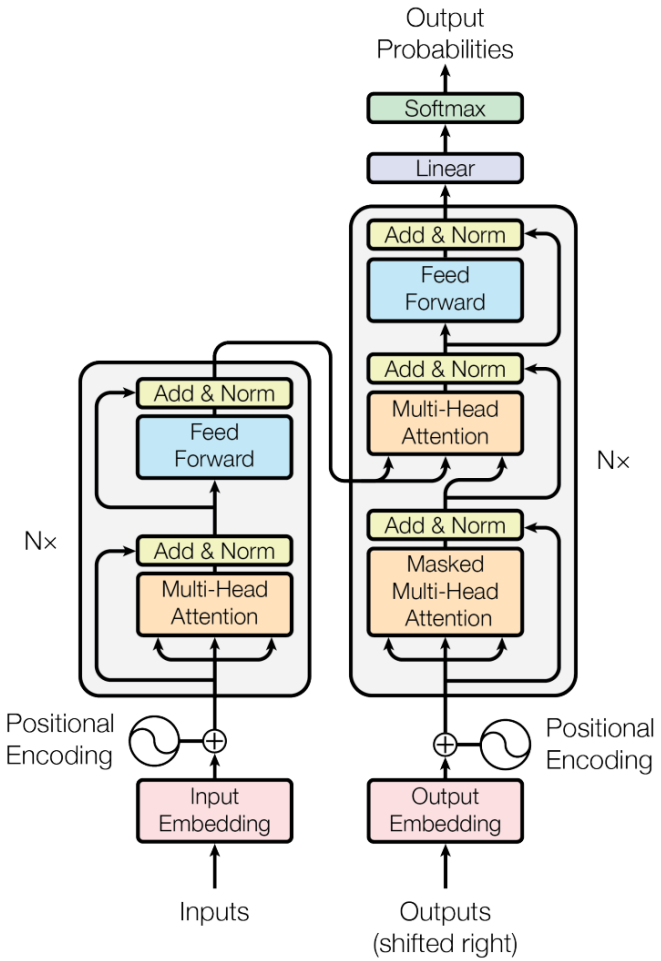
Generic neural MI estimators not tailored for finite-order TE

Why Transformers?

Attention captures long-range dependencies efficiently

Causal masking aligns with TE's finite history requirement

Scales better than RNNs for long contexts



TE vs. DI Rate

- TE: finite-order conditional MI (CMI)
- Directed Information (DI) rate: infinite memory CMI

$$\begin{aligned} \mathsf{I}((\mathbb{X} \to \mathbb{Y}) &:= \lim_{n\to\infty} \frac{1}{n} \mathsf{I}\left(X^n \to Y^n\right) \\ &= \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathsf{I}\left(X^i; Y_i | Y^{i-1}\right) \\ &= \lim_{n\to\infty} \mathsf{I}\left(X^n; Y_n | Y^{n-1}\right) \end{aligned}$$

- TE converges to DI rate under stationarity

What is TREET?

- Transformer-based estimator of TE using Donsker-Vardhan (DV) representation

> **Theorem 2 (DV representation)** *For any,* $P, Q \in \mathcal{P}(\mathcal{X})$*, we have*
>
> $$\mathsf{D}_{\mathsf{KL}}(P\|Q) = \sup_{f:\mathcal{X}\to\mathbb{R}} \mathbb{E}_P[f] - \log\left(\mathbb{E}_Q[e^f]\right),$$
>
> *where the supremum is taken over all measurable functions* f *with finite expectations.*

- Shared-weights dual potentials with tailored attention.
- Consistent for order-l TE.

DV Objective for TE

- TE can be expressed as difference of two KL divergences

Lemma 2 (TE as KL Divergences) *TE decomposes as*

$$\mathsf{TE}_{X\to Y}(l) = \mathsf{D}_{Y_l|Y^{l-1}X^l\|\widetilde{Y}_l} - \mathsf{D}_{Y_l|Y^{l-1}\|\widetilde{Y}_l},$$

where

$$\mathsf{D}_{Y_l|Y^{l-1}\|\widetilde{Y}_l} := \mathsf{D}_{\mathsf{KL}}\left(P_{Y_l|Y^{l-1}}\|\widetilde{P}_Y\middle|P_{Y^{l-1}}\right),$$

$$\mathsf{D}_{Y_l|Y^{l-1}X^l\|\widetilde{Y}_l} := \mathsf{D}_{\mathsf{KL}}\left(P_{Y_l|Y^{l-1}X^l}\|\widetilde{P}_Y\middle|P_{Y^{l-1}X^l}\right)$$

- Each term is **optimized via DV representation**, where the potential function is a transformer.
- Reference sampling (Gaussian / Uniform) enables stable estimation of KL divergence terms

Consistency Guarantee

- Under stationarity and ergodicity, TREET converges to true TE
- Proven for fixed memory parameter l.
- $\tilde{Y}$ is i.i.d. under absolutely continuous reference measure over the alphabet $\mathcal{Y}$.
- Each term is a lower bound estimator of the DKL in Lemma 2, respectively.

$$\widehat{\mathsf{TE}}_{X\to Y}(D_n;l) = \sup_{g_{xy}\in\mathcal{G}_{\text{ctf}}^{XY}} \widehat{\mathsf{D}}_{Y_l|Y^{l-1}X^l\|\widetilde{Y}_l}(D_n,g_{xy}) - \sup_{g_y\in\mathcal{G}_{\text{ctf}}^{Y}} \widehat{\mathsf{D}}_{Y_l|Y^{l-1}\|\widetilde{Y}_l}(D_n,g_y),$$

where

$$\widehat{\mathsf{D}}_{Y_l|Y^{l-1}\|\widetilde{Y}_l}(D_n,g_y) := \frac{1}{n}\sum_{i=1}^{n} g_y\left(Y_i^{i+l}\right) - \log\left(\frac{1}{n}\sum_{i=1}^{n} e^{g_y\left(\widetilde{Y}_{i+l},Y_i^{i+l-1}\right)}\right),$$

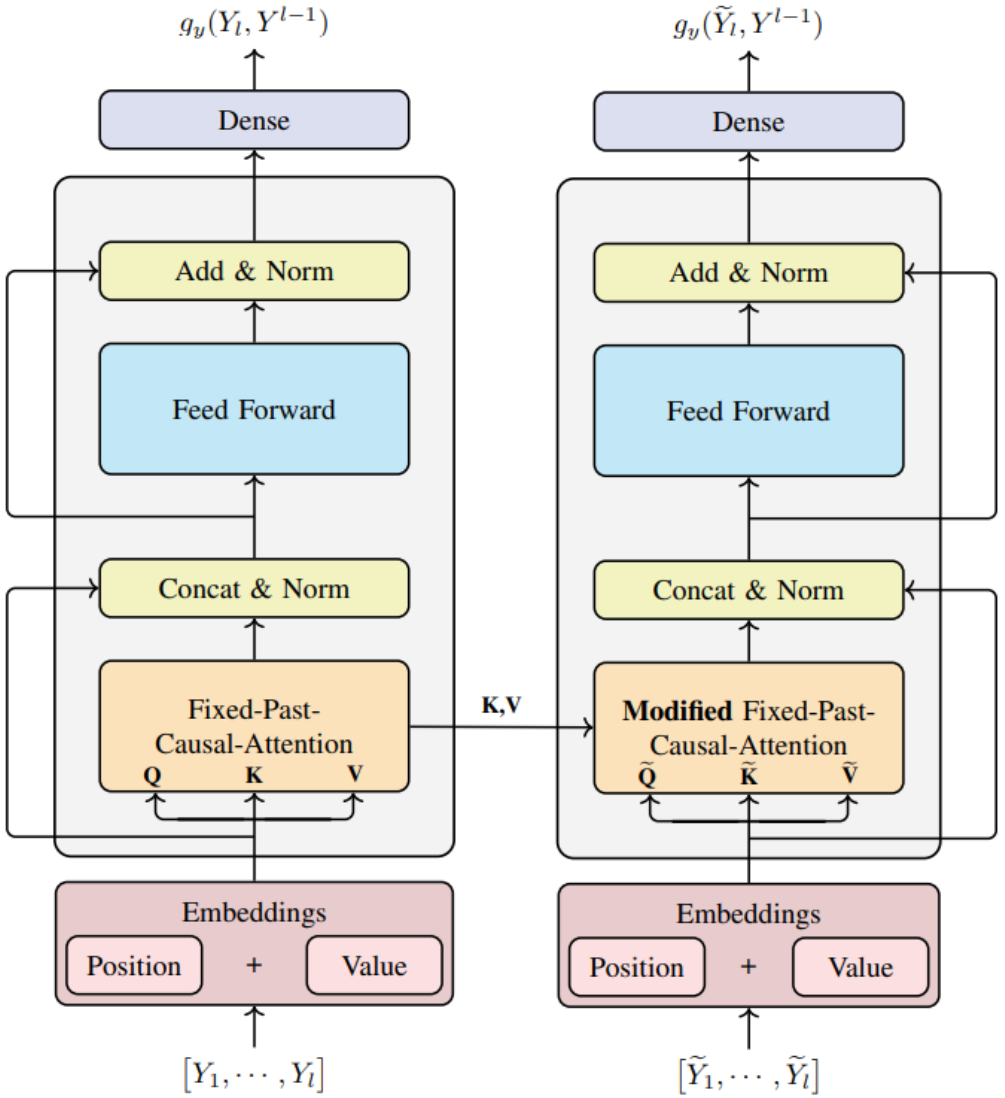
$$\widehat{\mathsf{D}}_{Y_l|Y^{l-1}X^l\|\widetilde{Y}_l}(D_n,g_{xy}) := \frac{1}{n}\sum_{i=1}^{n} g_{xy}\left(Y_i^{i+l},X_i^{i+l}\right) - \log\left(\frac{1}{n}\sum_{i=1}^{n} e^{g_{xy}\left(\widetilde{Y}_{i+l},Y_i^{i+l-1},X_i^{i+l}\right)}\right).$$

TREET Architecture

Two passes through the same transformer (shared weights)

Fix-Past-Causal-Attention (FPCA) for main term; modified FPCA for reference term – reuses past keys and values to simulate the same conditionals

Stable training and interpretable lag-wise attention



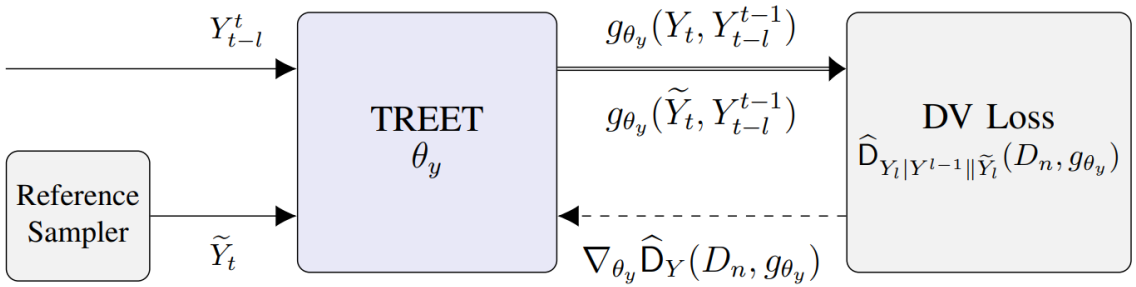
Training & Implementation

- Sampled order l sequences of $\mathbb{Y}$ and joint $\mathbb{X}, \mathbb{Y}$ process for each DKL term – unsupervised training for maximization of DV representation.
- Mini-batch optimization via gradient ascent. with reference sampling – tested on Uniform and Gaussian sampling
- FPCA and modified FPCA enforce causal structure to preserve no past beyond order l and future leakage. Parallel $(L - l)$ outputs for sequence length L.

Algorithm 1 TREET

Input: Joint process samples D_n; Observation length $l \in \mathbb{N}$.
Output: $\widehat{\mathsf{TE}}_{X \to Y}(D_n; l)$ - TE estimation.

1: NNs initialization g_{θ_y}, $g_{\theta_{xy}}$ with corresponding parameters θ_y, θ_{xy}.
2: **Step 1 – Optimization:**
3: **repeat**
4: Draw a batch B_m: $m < n$ sub-sequences, length $L > l$ from D_n, with reference samples $P_{\widetilde{Y}}$ for each.
5: Compute both potentials $\widehat{\mathsf{D}}_{Y_l|Y^{l-1}X^l\|\widetilde{Y}_l}(B_m, g_{\theta_{xy}})$, $\widehat{\mathsf{D}}_{Y_l|Y^{l-1}\|\widetilde{Y}_l}(B_m, g_{\theta_y})$ via (20).
6: Update parameters:
7: $\theta_{xy} \leftarrow \theta_{xy} + \nabla_{\theta_{xy}} \widehat{\mathsf{D}}_{Y_l|Y^{l-1}X^l\|\widetilde{Y}_l}(B_m, g_{\theta_{xy}})$
8: $\theta_y \leftarrow \theta_y + \nabla_{\theta_y} \widehat{\mathsf{D}}_{Y_l|Y^{l-1}\|\widetilde{Y}_l}(B_m, g_{\theta_y})$
9: **until** convergence criteria.
10: **Step 2 – Evaluation:** Evaluate for a sub-sequence (20) and (19a) to obtain $\widehat{\mathsf{TE}}_{X \to Y}(D_n; l)$.



TE Benchmark: Long-Memory Synthetic Data

- Compared TREET vs. TENE and Copnet – both ignoring time settings and using the sequence as a whole vector.
  - **TENE** – DV based TE estimator,

$$\mathrm{TE}_{X\to Y}(l) = \mathrm{I}(X^l; Y_l | Y^{l-1}) = \mathrm{I}(X^l; Y_l, Y^{l-1}) - \mathrm{I}(X^l; Y^{l-1})$$

  - **Copnet** – KNN based estimator – 4 coupla entropy estimators,

$$\mathrm{I}(X) = -\mathrm{h}_C(X);$$
$$\mathrm{TE}_{X\to Y}(l) = \mathrm{h}_C(Y_l, Y^{l-1}, X^l) + \mathrm{h}_C(Y_l, Y^{l-1}) + \mathrm{h}_C(Y^{l-1}, X^l) - \mathrm{h}_C(Y^{l-1})$$

- TREET robust to long context lengths (up to $l = 99$)

TE Benchmark: Long-Memory Synthetic Data

$$Y_t = \begin{cases} Z_t, & \text{if } Y_{t-1} < \lambda, \\ \rho X_{t-1} + \sqrt{1-\rho^2} Z_t, & \text{else,} \end{cases}$$

Process parameters:

$$\lambda \in \mathbb{R}, \rho = 0.9$$

For any order TE, $l \geq 1$, the value is constant since X is i.i.d. and Y is 1-order Markov.

λ		-3	-2	-1	0	1	2	3
Model, l \ Ground Truth		0.829	0.811	0.699	0.415	0.132	0.019	0.001
TREET	1	0.812	0.792	0.667	0.395	0.126	0.016	0.0
	2	0.826	0.796	0.681	0.392	0.117	0.014	0
	4	0.825	0.798	0.682	0.393	0.115	0.013	0
	7	0.82	0.799	0.679	0.405	0.121	0.015	0.001
	9	0.815	0.802	0.686	0.403	0.126	0.017	0.001
	19	0.824	0.805	0.69	0.405	0.128	0.017	0.001
	49	0.829	0.811	0.694	0.409	0.128	0.018	0.001
	99	0.829	0.81	0.693	0.41	0.115	0.017	0.001
TENE	1	0.823	0.807	0.696	0.416	0.126	0.014	0
	2	0.814	0.782	0.688	0.39	0.115	0.013	0
	4	0.43	0.76	0.602	0.382	0.119	0.013	-0.001
	7	>10	>10	0.698	0.354	0.09	0.013	0.0
	9	>10	>10	>10	0.359	0.091	0.014	0.0
	19	>10	>10	>10	1.796	0.038	0.021	0
	49	<-10.0	<-10.0	<-10.0	>10	0.027	0.01	-0.002
	99	>10	>10	>10	>10	0.102	-0.002	-0.002
Copnet	1	0.835	0.81	0.688	0.397	0.111	0.0	-0.022
	2	0.819	0.786	0.676	0.377	0.106	-0.004	-0.017
	4	0.9	0.864	0.747	0.469	0.193	0.087	0.079
	7	1.297	1.268	1.135	0.903	0.649	0.571	0.575
	9	1.703	1.679	1.558	1.357	1.106	1.054	1.053
	19	4.862	4.834	4.75	4.574	4.431	4.428	4.432
	49	>10	>10	>10	>10	>10	>10	>10
	99	>10	>10	>10	>10	>10	>10	>10

Channel Capacity Estimation

As stated before, TE converges to the DI rate, thus TREET can estimate the DI rate – which achieves the channel capacity (for optimized input distribution)

Remark 1 (Channel capacity) *Consider channels with and without feedback links from the channel output back to the encoder. The feedforward capacity of a channel sequence* $\left\{P_{Y^n \| X^n}\right\}$, *for* $n \in \mathbb{N}$ *is*

$$C_{\mathsf{FF}} = \lim_{n\to\infty} \sup_{P_{X^n}} \frac{1}{n} \mathsf{I}(X^n; Y^n),$$

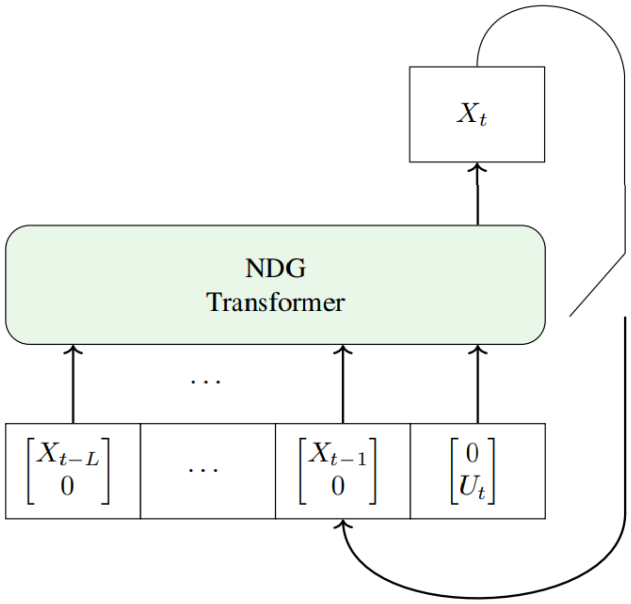
and the feedback capacity is

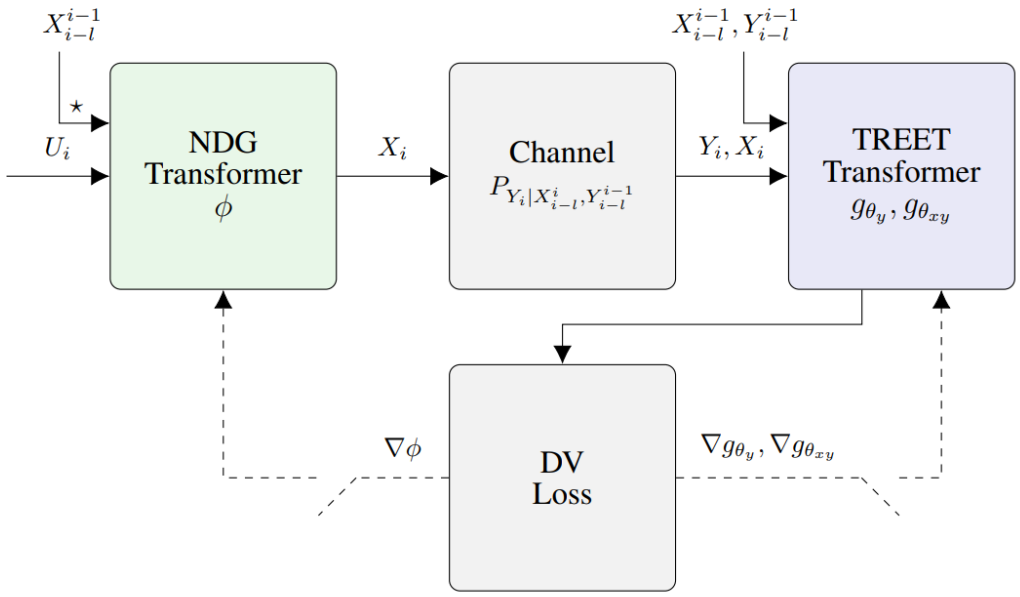
$$C_{\mathsf{FB}} = \lim_{n\to\infty} \sup_{P_{X^n \| Y^{n-1}}} \frac{1}{n} \mathsf{I}(X^n \to Y^n). \tag{29}$$

The achievability of the capacities is further discussed in [38], [39]. [34] showed that for non-feedback scenario, the optimization problem over $P_{X^n \| Y^n}$ *can be translated to* P_{X^n}

Channel Capacity Estimation

Neural Density Generator (NDG) learns input distribution maximizing TE – thus, TE is optimized and estimated in an alternate procedure.





Channel Capacity Estimation

- Applied to AWGN, AR(1), MA(1) channels with/without feedback.
- TREET matches theoretical capacities and the DI rate neural estimator - DINE

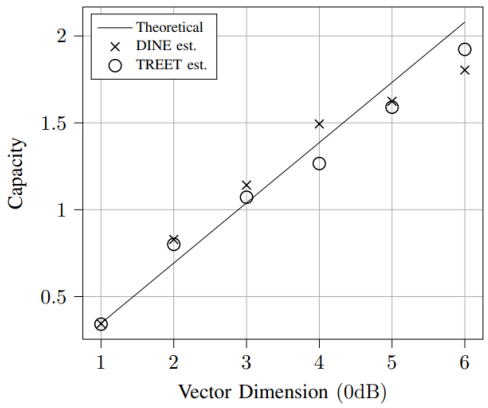
Algorithm 2 Continuous TREET optimization

Input: Continuous sequence-to-sequence system $\mathcal{S}$; Observation length $l \in \mathbb{N}$.

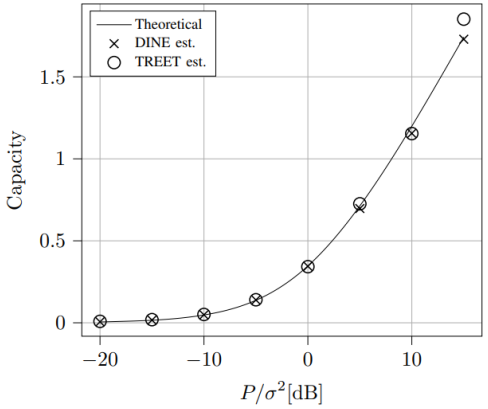
Output: $\widehat{\mathsf{TE}}^{\star}_{X\to Y}(U^n; l)$, optimized NDG.

1: NNs initialization g_{θ_y}, $g_{\theta_{xy}}$ and h_ϕ with corresponding parameters $\theta_y, \theta_{xy}, \phi$.
2: **repeat**
3: Draw noise U^m, $m < n$.
4: Compute batch B^ϕ_m sized m using NDG, $\mathcal{S}$
5: **if** training TREET **then**
6: Perform TREET optimization - Step 1 in Algorithm 1.
7: **else** Train NDG
8: Compute $\widehat{\mathsf{TE}}_{X\to Y}(B^\phi_m, g_{\theta_y}, g_{\theta_{xy}}, h_\phi; l)$ using (19a).
9: Update NDG parameters:
10: $\phi \leftarrow \phi + \nabla_\phi \widehat{\mathsf{TE}}_{X\to Y}(B^\phi_m, g_{\theta_y}, g_{\theta_{xy}}, h_\phi; l)$
11: **until** convergence criteria.
12: Draw U^m to produce l length sequence and evaluate $\widehat{\mathsf{TE}}_{X\to Y}(D^\phi_n; l)$.
13: **return** $\widehat{\mathsf{TE}}^{\star}_{X\to Y}(U^n; l)$, optimized NDG.

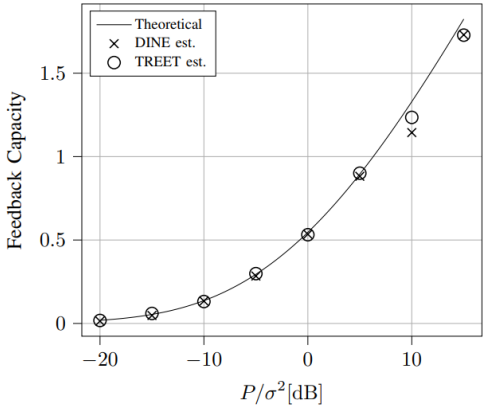
Channel Capacity Estimation



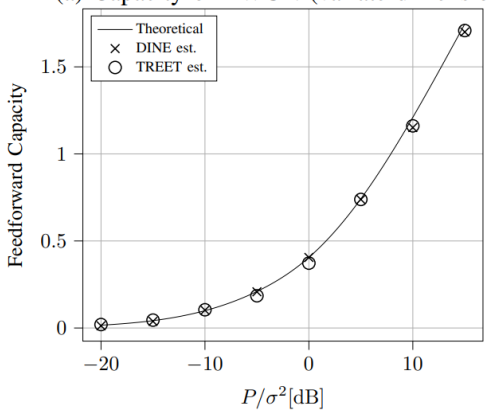
(a) Capacity of AWGN (variate dimension).



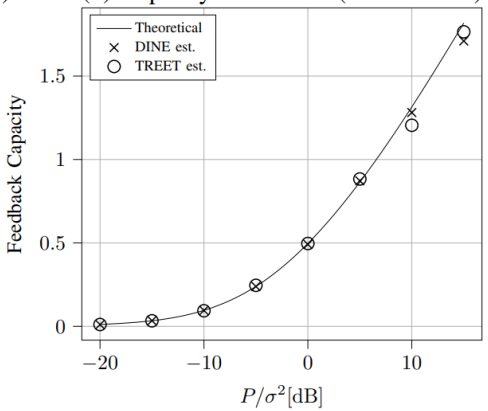
(b) Capacity of AWGN (variate SNR).



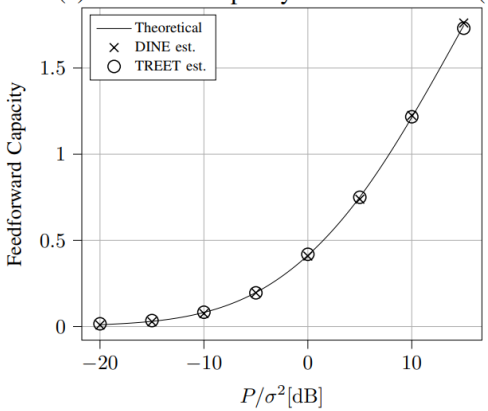
(c) Feedback Capacity of Gaussian MA(1).



(d) Feedforward Capacity of Gaussian MA(1).



(e) Feedback Capacity of Gaussian AR(1).



(f) Feedforward Capacity of Gaussian AR(1).

Channel Capacity Estimation For Long Memory Analysis

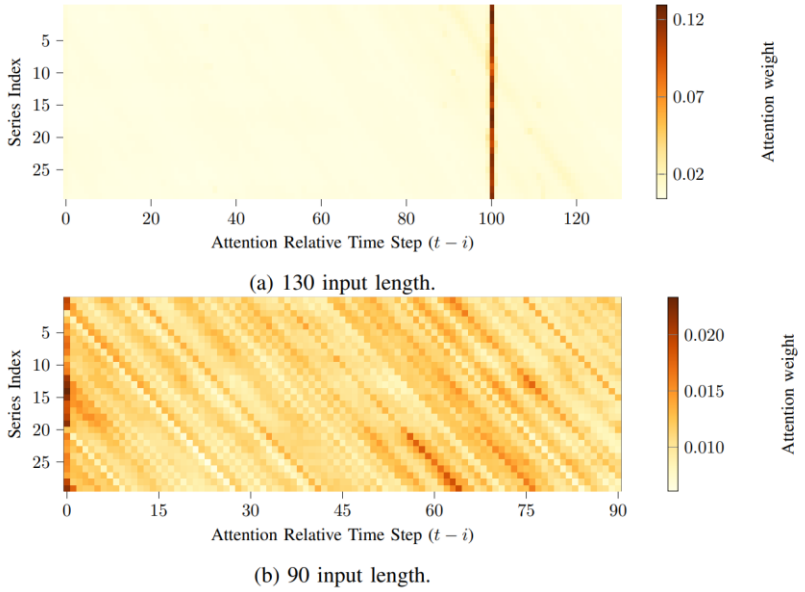
Process capacity estimation – GMA(100),

$$Z_t = N_t + \alpha N_{t-100},$$
$$Y_t = X_t + Z_t, \qquad t \in \mathbb{Z}.$$

FPCA visualization of influential lags for shorter or larger memory than the process' order

l	TREET		DINE	
	Estimated capacity [nat]	Absolute error (%)	Estimated capacity [nat]	Absolute error (%)
60	0.19	53	**0.30**	**25**
70	0.29	29	**0.35**	**15**
80	0.28	31	**0.34**	**17**
90	0.29	28	**0.30**	**26**
100	**0.37**	**9**	0.36	12
110	**0.38**	**7**	0.33	20
120	**0.36**	**11**	0.33	18
130	**0.36**	**11**	0.34	17
140	**0.35**	**14**	0.26	35

Stable Estimation



True capacity = 0.405 [nat]

Density Estimation via TREET

- Optimized potentials yield the following log-likelihood ratio (LLR) –

$$f^{\star}_{y,l} := \log\left(\frac{dP_{Y^l}}{d(P_{X^lY^{l-1}} \otimes \widetilde{P}_{Y_l})}\right) = \log p_{Y_l|X^lY^{l-1}} - \log \widetilde{p}_{Y_l}$$

$$f^{\star}_{y,l} := \log\left(\frac{dP_{Y^l}}{d(P_{Y^{l-1}} \otimes \widetilde{P}_{Y_l})}\right) = \log p_{Y_l|Y^{l-1}} - \log \widetilde{p}_{Y_l}$$

- The conditional density estimator can be derived out-of-the-box –

$$P_{Y_t|Y^{t-1}}(y_t|y^{t-1}) \approx \exp\left(\widehat{\mathsf{D}}_{Y_l|Y^{l-1}\|\widetilde{Y}_l}(D_n, g_y)\right) \cdot \widetilde{P}_{Y_t}(y_t)$$

- Entire distribution derived from normalized values from grid input of $\{y_t\} \subset \mathcal{Y}$.
- Competitive with MDN, KDE, Kalman on HMM-like processes and handles long delays and noise models robustly

Same can be derived for causally conditioned on X process

Density Estimation via TREET

$$X_t = \alpha X_{t-1} + \beta X_{t-k} + W_t,$$
$$Y_t = \gamma X_t + V_t,$$

Where W_t, V_t are i.i.d noise, $\mathcal{N}(0,1)$ or $U[-1,1]$.

Model	Gaussian		Uniform	
	KLD	TV	KLD	TV
Kalman	1.076	0.467	1.304	0.408
KDE	1.135	0.608	0.847	0.417
MDN	1.098	0.632	0.667	0.460
DINE	**0.795**	0.525	**0.475**	**0.291**
TREET	0.797	**0.524**	0.482	0.296

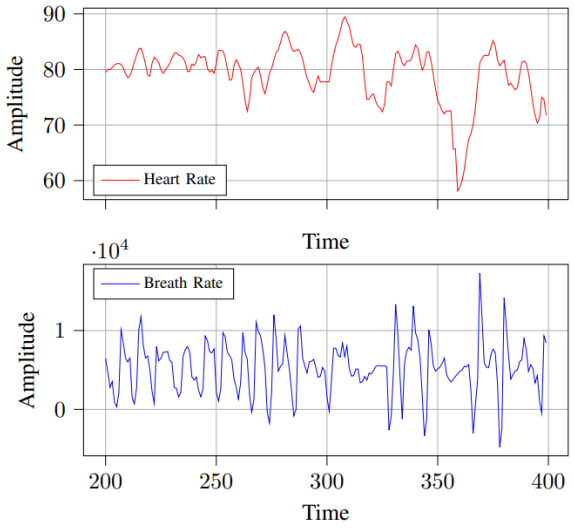
1-order process - $\beta = 0$

k	TREET		DINE		MDN		Kalman	
	KL	TV	KL	TV	KL	TV	KL	TV
2	0.933	0.569	0.934	0.570	1.460	0.708	**0.636**	**0.405**
5	1.036	0.601	**0.875**	**0.556**	1.454	0.707	1.284	0.599
10	**0.984**	**0.585**	1.299	0.662	1.450	0.706	1.239	0.603
15	**0.992**	**0.587**	1.441	0.704	1.448	0.707	1.232	0.602
25	**0.993**	**0.587**	1.451	0.706	1.443	0.704	1.196	0.598
50	**0.993**	**0.589**	1.452	0.707	1.453	0.707	1.193	0.597

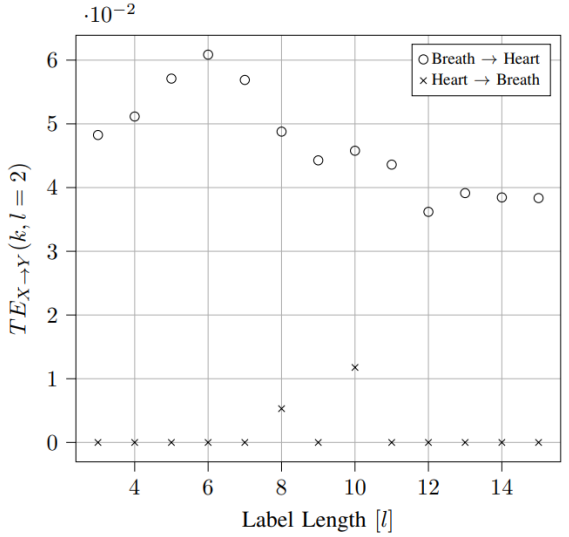
k-order process - $\beta \neq 0$ and $k > 1$

Apnea Case Study: Feature-Level Causal Analysis

- Analyzed **respiration** ⇄ **heart-rate** influence in sleep apnea patients
- TE directionality aligns with clinical understanding – breathing affects heart rate
- Different orders of TE estimation reveals causal relationships



(a) Sampled sequence of Apnea dataset.



(b) TE estimation for variable length history of Y.

Future Scope: TREET Time-Series Applications

Cross-Domain Potential
Apply TREET across diverse fields such as:

Communications – analyzing feedback and memory in channels

Physiology – uncovering causal relationships in biomedical signals

Finance – detecting directional influence in market dynamics

Advanced Applications of TREET

TREET enables powerful time-series analysis across several domains, including:

Feature Selection: Identifying causally relevant variables in complex datasets.

Anomaly Detection: Detecting irregularities in single or joint processes.

Control & Decision Systems: Enhancing decision-making through causal insights.

Future Scope: Information Theory & Deep Learning Application

1. Diffusion Models for Improved KLD and MI Estimation:

- Current estimators struggle with high dimensional KLD and MI
- Diffusion models can break down each variational representation task (DV, MINE, NWJ, InfoNCE) to smaller ones

$$\mathsf{T}^{\star}_{\text{sum}}(\mathbb{P}_0, \mathbb{P}_K) := \sum_{k=0}^{K-1} \mathsf{T}_k(\mathbb{P}_k \parallel \mathbb{P}_{k+1})$$

$$\begin{aligned}\mathsf{I}_{\text{VR,diffused}}(X;Y) &:= \frac{1}{N} \sum_{(X_i,Y_i)_{i=1}^N \sim \mathbb{P}_{XY}} \mathsf{T}^{\star}_{\text{sum}}(X_i, Y_i) \\ &\overset{(a)}{=} \mathbb{E}_{\mathbb{P}_{XY}}\left[\log \frac{\mathbb{P}_{XY}(X,Y)}{\mathbb{P}_X(X)\mathbb{P}_Y(Y)}\right] \\ &= \mathsf{I}(X;Y)\end{aligned}$$

Future Scope: Information Theory & Deep Learning Application

1. Diffusion Models for Improved Mutual Information Estimation:

- Proved lower bound for estimating the KLD, from optimal set of functions $\{T_k^\star\}_{k=0}^{K-1}$

Bernoulli Diffusion Process: Define $\mathbb{P}_k$ as a probabilistic mixture between $\mathbb{P}_0$ and $\mathbb{P}_K$, controlled by the parameter α_k:

$$\mathbb{P}_k = (1 - \alpha_k)\mathbb{P}_0 + \alpha_k \mathbb{P}_K.$$

This process results in a stochastic switching between samples from $\mathbb{P}_0$ and $\mathbb{P}_K$:

$$X_k = \begin{cases} X_0, & \text{with probability } 1 - \alpha_k, \\ X_K, & \text{with probability } \alpha_k. \end{cases}$$

Lemma (D_{KL} **upper bound - Bernoulli)**

$$\mathsf{D}_{\mathsf{KL}}(P_0||P_K) \geq \sum_{k=0}^{K-1} \mathsf{D}_{\mathsf{KL}}(P_k||P_{k+1}).$$

Future Scope: Information Theory & Deep Learning Application

2. Optimizing Inference of Discrete Diffusion Language Models Through MI:

- Diffusion models, discrete ones in particular, suffer from many function evaluations until resulting with a good quality generated content

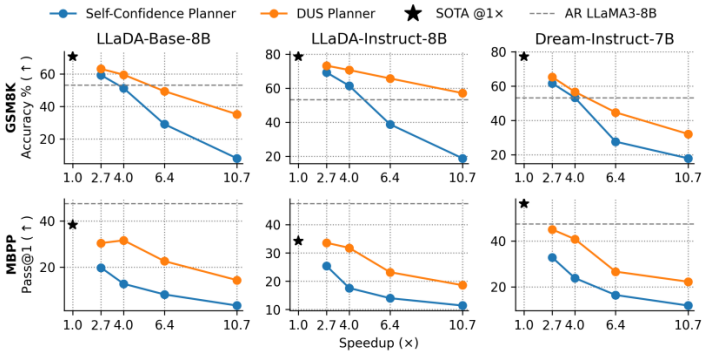
- Dilated Unmasking Scheduler (DUS) - Assuming a Markov model, uncovering tokens at the same time can be done with minimal shared information between them, thus generation preserves its' quality

Lemma 1 *Under a fast-mixing first-order Markov chain, let* $i_1 < \cdots < i_k$ *be the indices selected by DUS. Then*

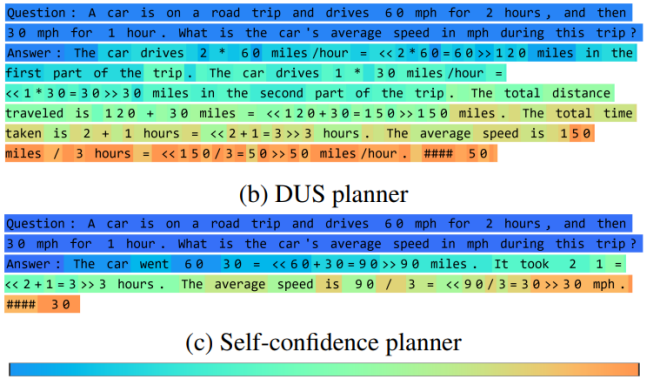
$$H(X_{i_1}, \ldots, X_{i_k} \mid \mathcal{S}_t) \approx \sum_{j=1}^{k} H(X_{i_j} \mid \mathcal{S}_t). \qquad (8)$$

Future Scope: Information Theory & Deep Learning Application

2. Optimizing Inference of Discrete Diffusion Language Models Through MI:



(a) Score vs. speedup - DUS and self-confidence MDLM planners.



(b) DUS planner

(c) Self-confidence planner

Generation Start — Generation End

	Model	B=8 ×2.7		B=16 ×4		B=32 ×6.4		B=64 ×10.7	
		Conf.	DUS	Conf.	DUS	Conf.	DUS	Conf.	DUS
GSM8K	LLaDA–Base	59.29	**63.08**	51.23	**59.51**	29.04	**49.36**	8.04	**35.18**
	LLaDA–Instruct	69.22	**73.24**	61.41	**70.66**	38.74	**65.73**	18.73	**57.09**
	Dream–Instruct	61.64	**65.28**	53.22	**56.63**	27.60	**44.66**	17.89	**32.07**
MATH500	LLaDA–Base	16.6	**21.4**	11.2	**19.2**	6.0	**13.6**	2.6	**10.2**
	LLaDA–Instruct	21.4	**23.8**	15.4	**22.8**	10.8	**19.2**	8.0	**14.8**
	Dream–Instruct	22.4	**27.0**	15.4	**19.8**	7.2	**13.2**	4.0	**11.6**
Humaneval	LLaDA–Base	15.85	**25.61**	12.8	**19.51**	4.88	**14.02**	4.88	**6.71**
	LLaDA–Instruct	21.95	**28.05**	14.02	**23.17**	9.76	**10.37**	10.98	**11.59**
	Dream–Instruct	8.54	**14.63**	5.49	**11.59**	**6.71**	**6.71**	6.10	**9.15**
	DiffuCoder–Base	17.07	**28.66**	6.71	**38.41**	2.44	**21.95**	0.61	**6.10**
	DiffuCoder–Instruct	7.93	**22.56**	14.02	**20.12**	**13.41**	12.80	11.59	**8.54**
MBPP	LLaDA–Base	19.8	**30.4**	12.8	**31.6**	8.2	**22.6**	3.4	**14.4**
	LLaDA–Instruct	25.4	**33.6**	17.6	**31.8**	14.0	**23.2**	11.4	**18.6**
	Dream–Instruct	32.8	**45.0**	23.8	**40.8**	16.4	**26.6**	11.8	**22.2**
	DiffuCoder–Base	29.2	**48.6**	17.4	**43.0**	10.2	**27.4**	3.4	**17.2**
	DiffuCoder–Instruct	31.8	**46.4**	25.6	**43.6**	21.0	**26.6**	13.0	**18.2**

Thank you!

- TREET: TRansfer Entropy Estimation via Transformers, IEEE Access, 2025 vol. 13, pp. 126477-126495, 2025
  Omer Luxembourg, Dor Tsur, Haim Permuter
- Plan for Speed: Dilated Scheduling for Masked Diffusion Language Models, arXiv preprint arXiv:2506.19037, 2025
  Omer Luxembourg, Haim Permuter, Eliya Nachmani