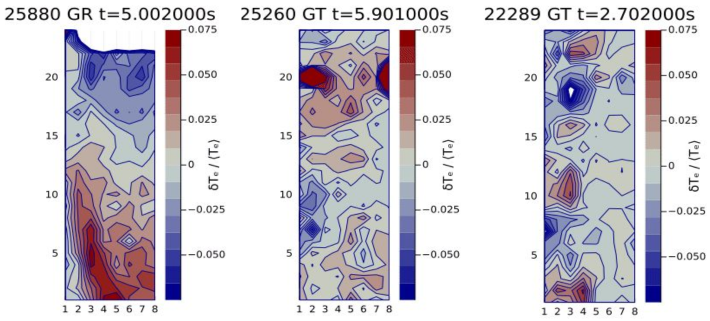


Unsupervised classification of KSTAR ECEI data

R. Kube[1], R.M. Churchill[1], J. Choi[2], M. Choi[3]

1 - Princeton Plasma Physics Laboratory, Princeton NJ 08540, USA

2 - Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

3 - Korea Institute for Fusion Energy, Daejeon 34133, Republic of Korea

This work was supported by R&D Programs of "KSTAR Experimental Collaboration and Fusion Plasma Research (EN2201-13)"

Fusion experiments produce ever larger data sets. Deep learning keeps scaling with dataset size.

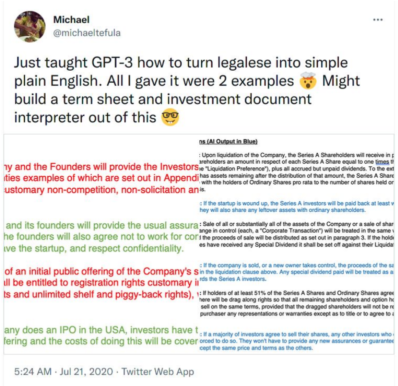
Data production rates at fusion experiments are ever increasing:

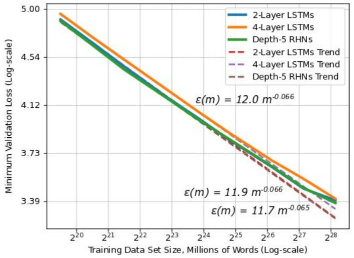
- ITER expected to produce ~2PB per day in full operation.
- KSTAR - order of TB/day
- NSTX-U - order of 1-10 GB/day

Today, a large fraction of measurement data is only written, but never analyzed for various reasons.

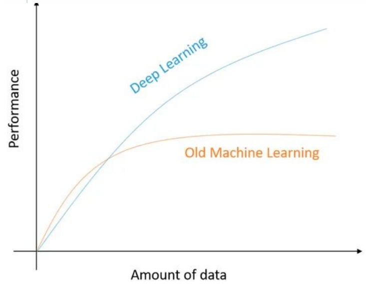
Automatic data labelling and extraction will be beneficial for long-pulse operation.

GPT-3 / Dall-e 2: O(10TB) training data sets





http://research.baidu.com



Alom et al. Electronics 8(3) 292 2019

Unsupervised learning promises to unlock deep learnings performance scaling with dataset size.

About 1 week of data labelling

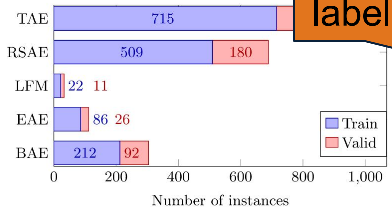


Figure 3. Number of instances for each of the manually-labeled AE types, for a random assignment of 450 training and 150 validation discharges. The distribution is heavily skewed towards RSAE and TAE activity.

Jalalvand et al. [NF **62** 026007 (2022)]

Manually annotating 500 spectrogram images

We then had to annotate, i.e. create the ground truth of all the 500 spectrograms to identify the modes activity and train the neural network. The annotation process was done manually using any simple picture editor, freely available in all operat-

Bustos et al. [PPCF 63 095002 (2021)]

10_000 shots labelled

Extended Data Table 2 | Datasets used h

Machine	Shot range	Total shots	disruptive	Note
JET train	66027 – 79853	2894	215	JET carbon wall campaigns C23-C27b
JET validate	66027 – 79853	1425	87	JET carbon wall campaigns C23-C27b
JET test	81852 – 83793	1191	174	JET ITER-like wall campaigns C28-30
DIII-D train	125500 – 168555	1734	407	DIII-D shots since 2006
DIII-D validate	125500 – 168555	853	197	DIII-D shots since 2006
DIII-D test	125500 – 168555	862	206	DIII-D shots since 2006

Shots were obtained from the respective machines. All shots that contain data for all signals were used. No shots were discarded for bad or abnormal data, or for being known testing shots or intentional disruptions.

Kates-Harbeck et al. [Nature 568 526 (2019)]

Databases of physics events have been used in various fusion research applications, including the development of scaling laws and disruption avoidance algorithms, yet they can be time-consuming and tedious to construct. This paper presents a novel application of the label spreading semi-supervised learning algorithm to accelerate this process by detecting distinct events in a large dataset of discharges, given few manually labeled examples. A high detection

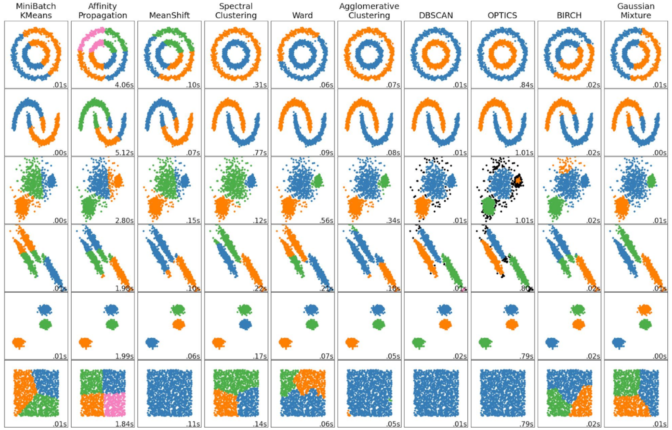
Montes et al. [NF 61 026022 (2021)]

Outline

1. Unsupervised Machine Learning
2. Dataset
3. Approach 1: Generative Adversarial Networks
4. Approach 2: Deep Divergence-based Clustering
5. Take-away and outlook

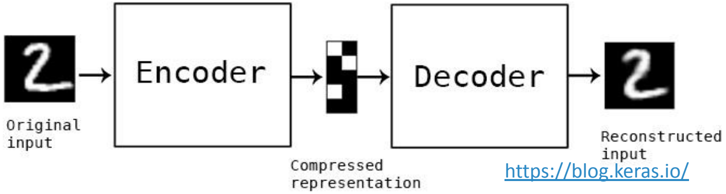
Unsupervised ML models identify structure within datasets and cluster it

Premise: A dataset contains some structure that allows to categorize it into separate groups - Input distribution p(x) contains information so that one can calculate p(y=k|x) where k is a class label

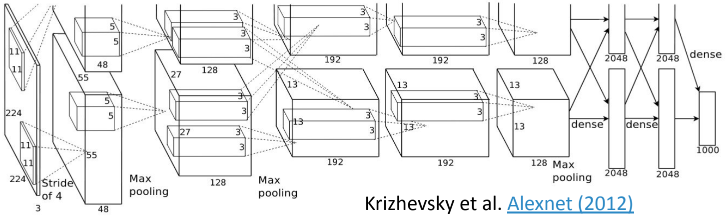


https://scikit-learn.org

- K-Means or DBSCAN have computational (and memory) complexity of $O(n^2)$
- Relying on Euclidian distance, unsuitable for images



https://blog.keras.io/



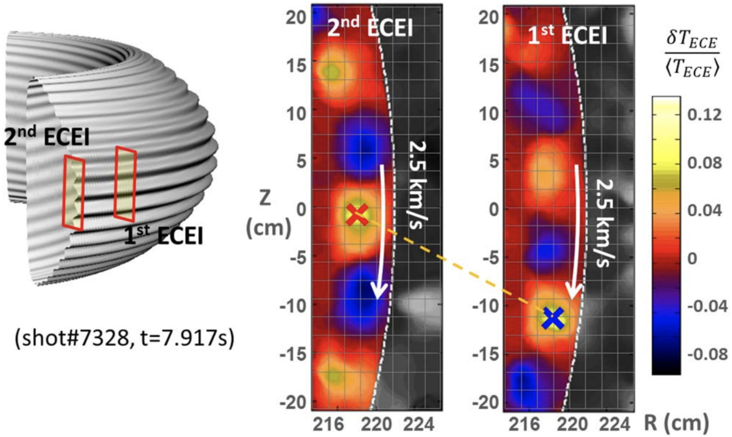
Krizhevsky et al. Alexnet (2012)

Integrating deep neural networks with clustering approaches in an end-to-end model is a novel direction of research.
Can such models classify ECEI data?

Outline

1. Unsupervised Machine Learning
2. Dataset
3. Approach 1: Generative Adversarial Networks
4. Approach 2: Deep Divergence-based Clustering
5. Take-away and Outlook

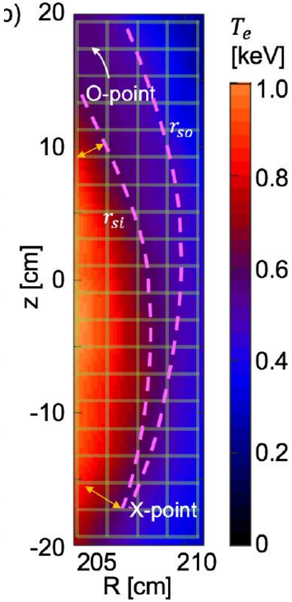
ECEI diagnostic at KSTAR used to study MHD phenomena

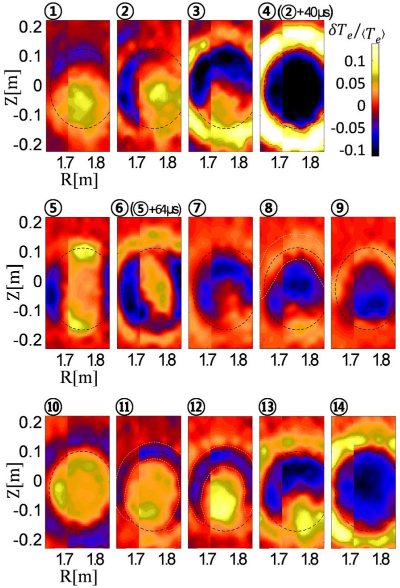


Yun et al. RSI 85 11D820 (2014)

- KSTAR ECEI diagnostic: samples 24x8=192 channels with ~MHz sampling rate
- Diagnostics produces image time-series with about 1GB/sec

Fast/Slow sawtooth crash Choe et al. NF 106038 2018





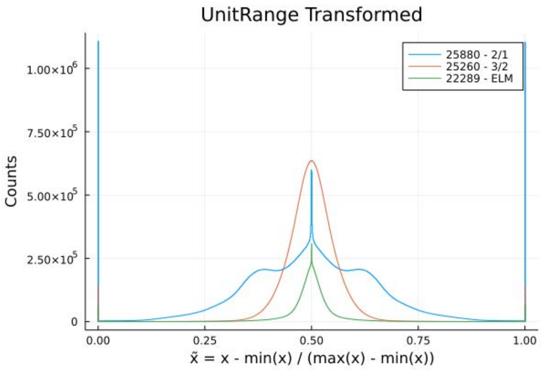
Interaction between MI and turbulence Choi et al NF 57 126058 2017

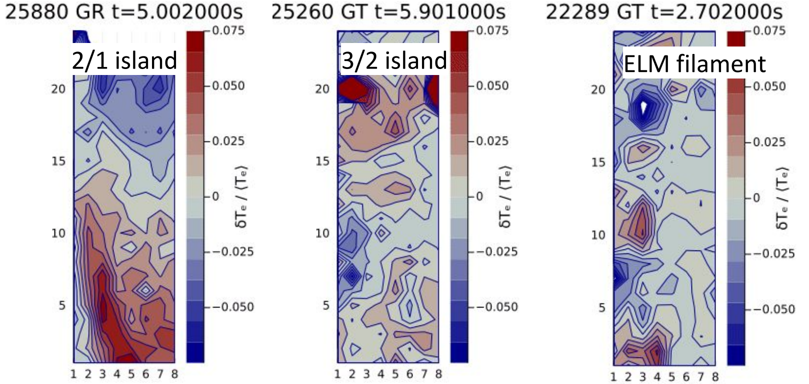
Overview: H. Park APX 4 2019

A dataset of MHD phenomena observed by ECEI is compiled

18 KSTAR Shots. 5 Shots with 3/2 magnetic islands, 12 shots with 2/1 magnetic islands, 1 Shot with ELM filaments. Varying lengths, about 0.5 - 5 seconds.

- Preprocessing: Applying a frequency filter, normalized and clamped to ±0.15 $\delta T_e / \langle T_e \rangle$



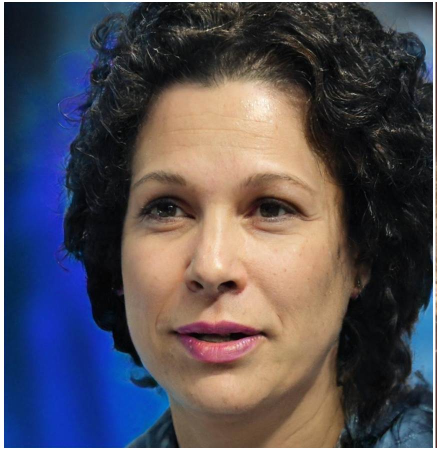
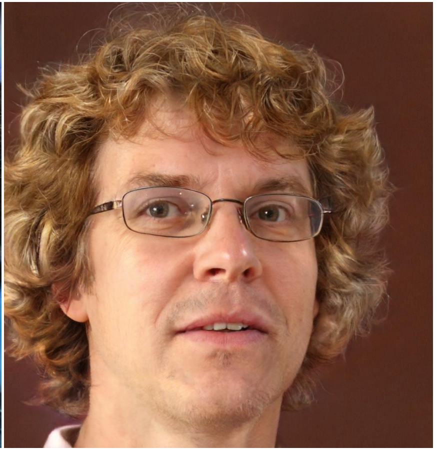


- Data understood as sequences - Patches are correlated across frames.
- Data is normalized to values between ±1
- Histograms are uni-modal, vary in structure, but are all symmetric.
- Bad channels may cause outliers - Need to be manually imputed
- Data from other class representatives are qualitatively similar

Outline

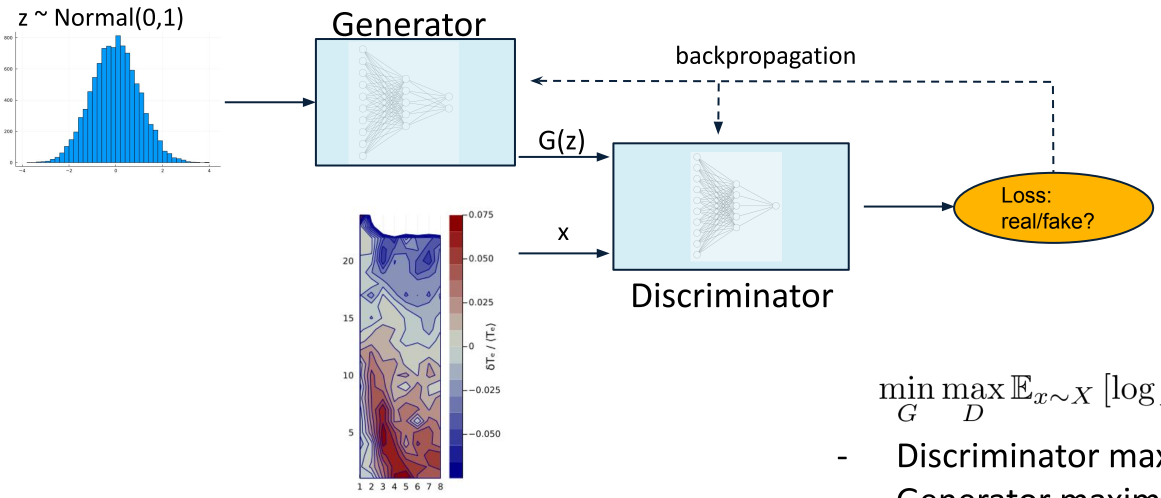
1. Unsupervised Machine Learning
2. Dataset
3. Approach 1: Generative Adversarial Networks
4. Approach 2: Deep Divergence-based Clustering
5. Take-away and Outlook

Generative Adversarial Networks allow point-wise sampling of arbitrary data distributions

StyleGAN2 - Karras et al. (2019) https://arxiv.org/abs/1912.04958 https://thispersondoesnotexist.com/

GANs consists of two models trained to out-smart the other one.



Goodfellow et al. (2014) https://arxiv.org/abs/1406.2661

https://arxiv.org › stat
[1406.2661] Generative Adversarial Networks - arXiv
by IJ Goodfellow · 2014 · Cited by 43701 — Generative Adversarial Networks. Authors:Ian J.

$$\min_G \max_D \mathbb{E}_{x\sim X}\left[\log p(y=1|x,D)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log 1 - p(y=1|G(z),D)\right.$$

- Discriminator maximizes log p(y=1|x,D)
- Generator maximizes p(y=1|G(z), D), which minimizes log(1-p(y=1|G(z), D))

Connection to classification: The Discriminator predicts a probability that a sample is from the distribution X.

Adapting the GAN loss function allows to classify images

- In standard GANs, D is a binary classifier trained to detect features that are not captured by G. These may not be the optimal features to perform classification.
- Instead, task D to assign class probabilities (k=1…K) for each sample
- G has to learn real features for the classes and D needs to distinguish these from real class features

$$p(y = k|x, D) = \frac{\exp D_k(x)}{\sum_{k=1}^{K} \exp D_k(x)}$$

CatGAN, Springenberg ICLR 2016

Discriminator:

1. Prior on class distribution: Use all classes equally
2. Certain about class assignment from X
3. Uncertain about class assignment from G(z)

$$\max_D H_X\left[p(y|D)\right] - \mathbb{E}_{x\sim X}\left[H(p(y|x, D))\right] + \mathbb{E}_{z\sim p_z(z)}\left[H(p(y|G(z), D))\right)$$

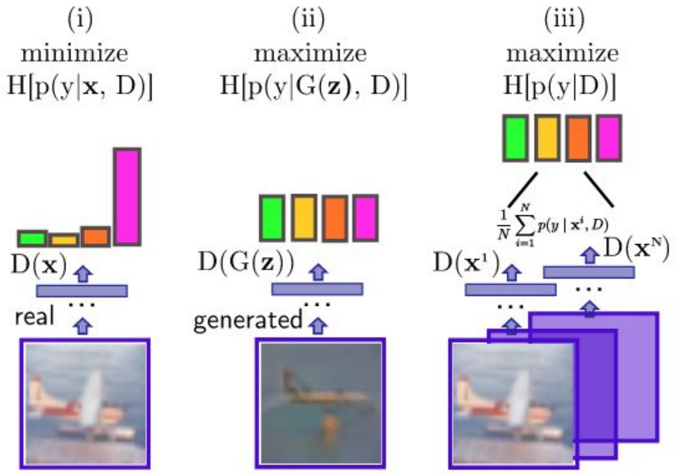
- Loss function can be extended to include cross-entropy in case labels are available.

Generator:

1. Use all classes equally
2. Generate examples with peaked class probability

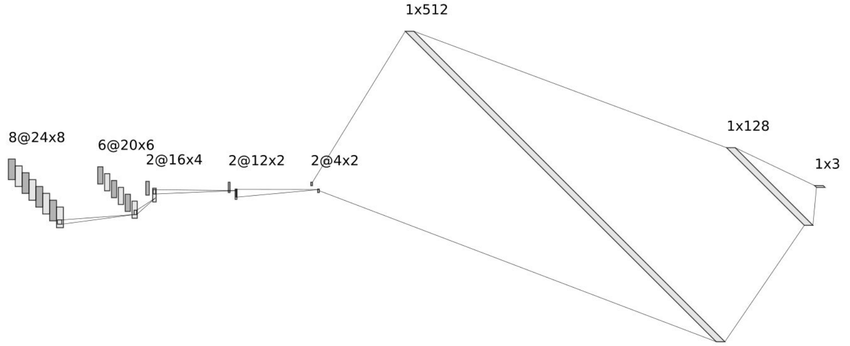
$$\min_G -H_X\left[p(y|D)\right] + \mathbb{E}_{z\sim p_z(z)}\left[H(p(y|G(z), D))\right)$$

H: Shannon entropy



Shallow convolutional architectures suitable for ECEI data

Discriminator performs series of 3d convolution
Input are 8 consecutive ECEI frames
Output is a 3-element vector, giving class probability

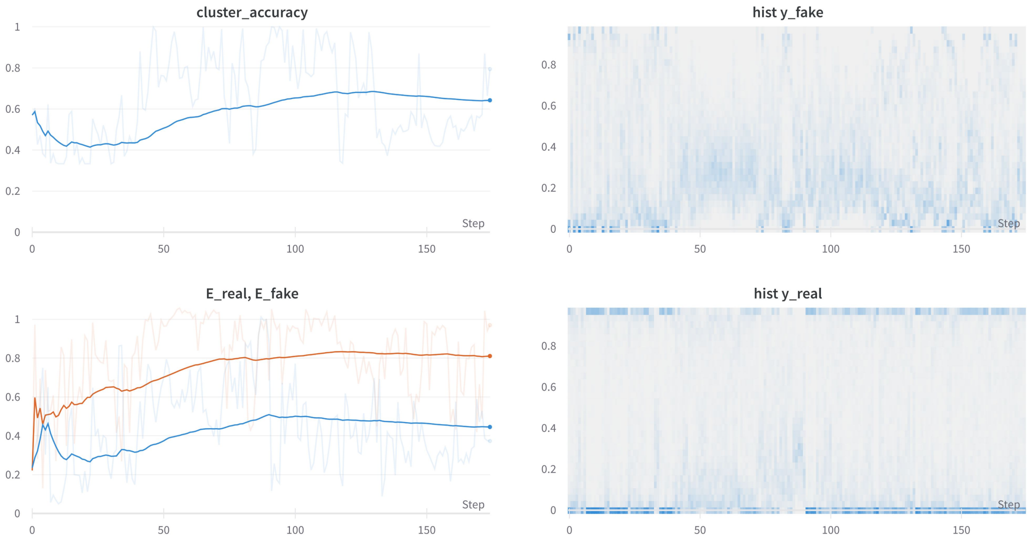


Filters: [5,3,3] (16ch) - [5,3,5] (16ch) - [5,3,1] (32ch) - [9,1,1](32ch)
BatchNorm applied after every convolutional layer
Minibatch Discrimination aids in avoiding mode collapse in the Generator
Salimans NeurIPS (2016)

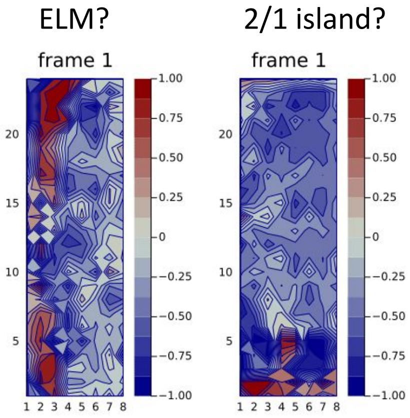
Generator
Dense(latent, 768, relu)
ConvTranspose((5,5,5), 32->32, stride=2, leakyrelu)
Conv((5,3,3), 32->32, pad=1, leakyrelu)
Conv((5,3,3), 32->16, pad=1, leakyrelu)
ConvTranspose((4,4,4), 16->16, stride=2, leakyrelu)
Conv((3,3,3), 16->16, pad=1, leakyrelu)
Conv((3,3,3), 16->1, tanh)

BatchNorm applied after every convolutional layer

No single metric tells whether the classifier is well trained

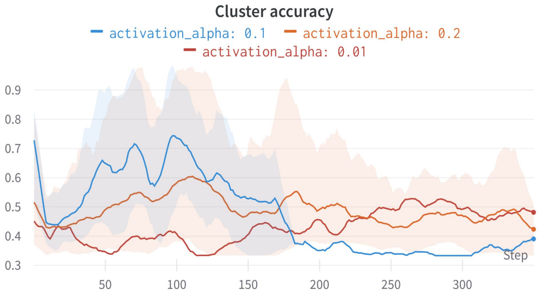


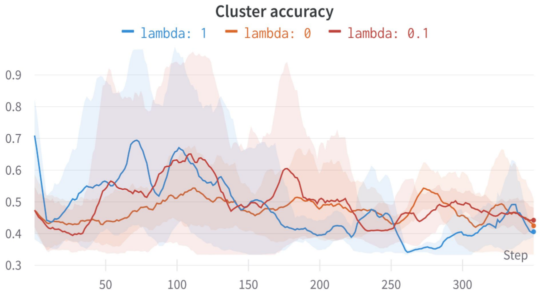
- Goal: Cluster with high accuracy
- Objective: E[H(p(y|G(z),D))] > E[H(p(y|x,D))]
  - More uncertainty for generated data than for real data
- Well trained GANs produce sequences similar to those in the training data

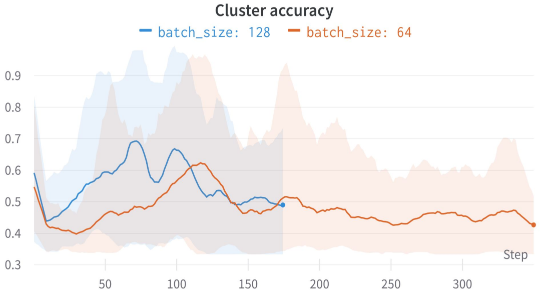


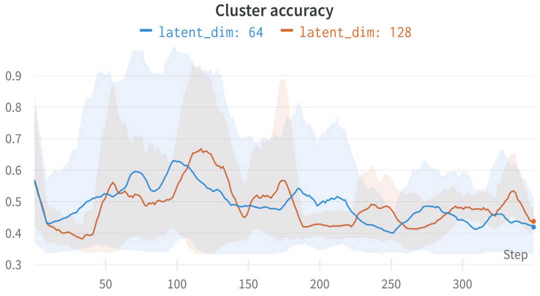
Evaluated using 0.1s (12,500 frames) per discharge and 3 discharges (2/1 Island, 3/2 island, ELM)
Parameters scanned:
Batch size: 64, 128. Learning rates: ((2e-4, 1e-3), (5e-4, 1e-3), (1e-3, 1e-3)), Channels: [16,16,32,32], [32,32,64,64], Lambda: (0.0, 0.1, 1.0), latent dim: 64, 128, alpha: 0.01, 0.1, 0.2 = 216 configurations.

GANs can cluster ECEi data reasonably well but are difficult to train









Reference implementation:

- 1.39 ± 0.28 error rate on MNIST
- 19.58 ± 0.58 error rate on CIFAR10 (semi-supervised)

Springenberg [ICLR 2016]

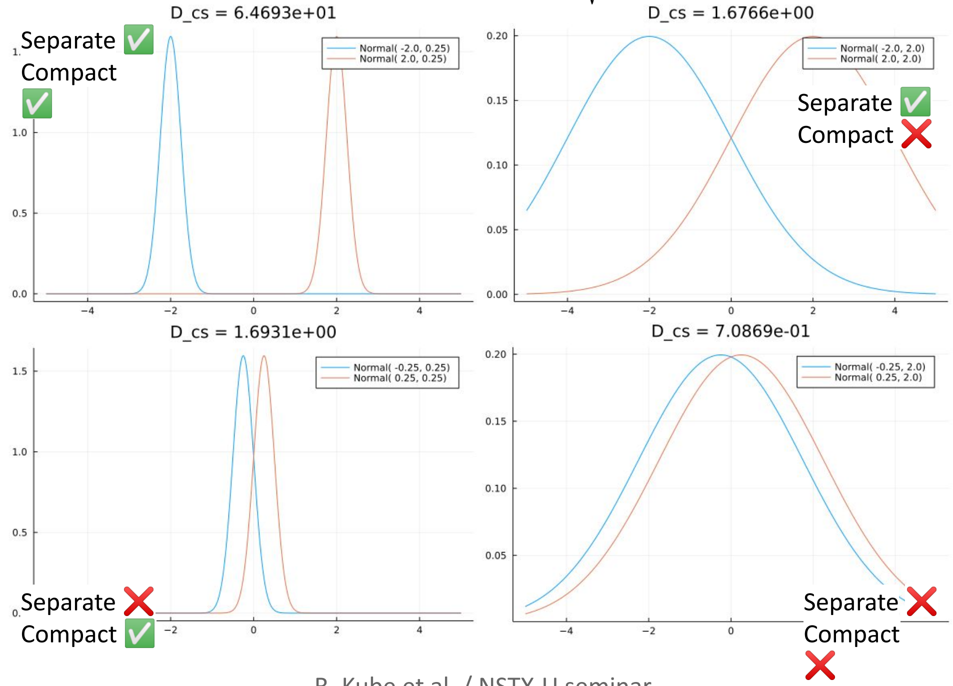
Outline

1. Unsupervised Machine Learning
2. Dataset
3. Approach 1: Generative Adversarial Networks
4. Approach 2: Deep Divergence-based Clustering
5. Take-away and Outlook

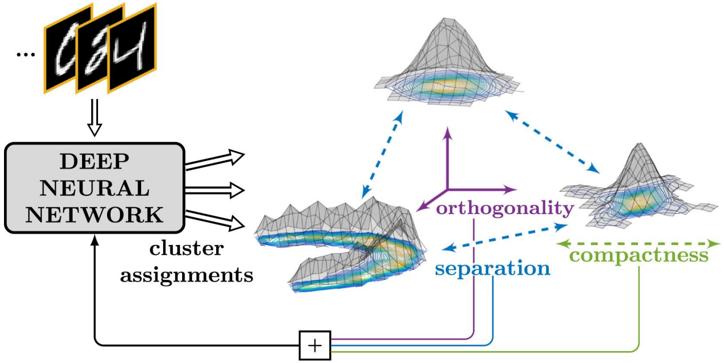
Divergences measure separation and compactness of distribution functions

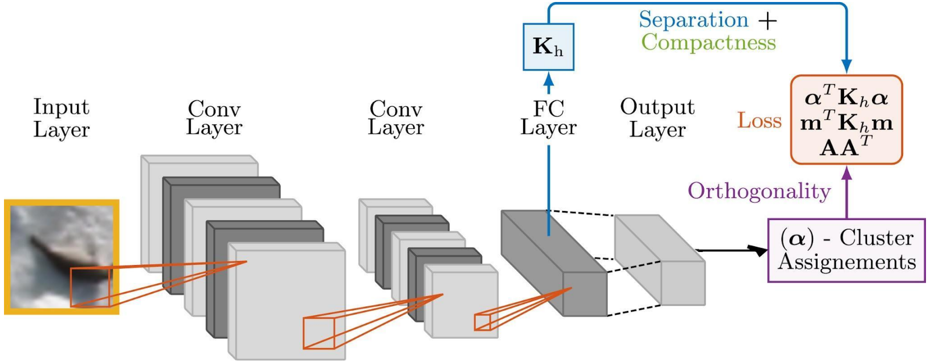
Cauchy-Schwarz Divergence

$$D_{\mathrm{CS}}\left(p_1(x), \ldots, p_K(x)\right) = -\log \frac{1}{K} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K-1} \frac{\int p_i(x) p_j(x) \mathrm{d}(x)}{\sqrt{\int p_i(x)^2 \mathrm{d}(x) \int p_j(x)^2 \mathrm{d}(x)}}$$



Integrate divergence-based clustering and deep neural networks.





Deep Neural Network extracts features from input.
Loss function enforces separation and compactness of resulting cluster.
Softmax output is a k-simplex in R^k. Orthogonality of cluster assignments can be optimized by exploiting the geometry of the output spaced.

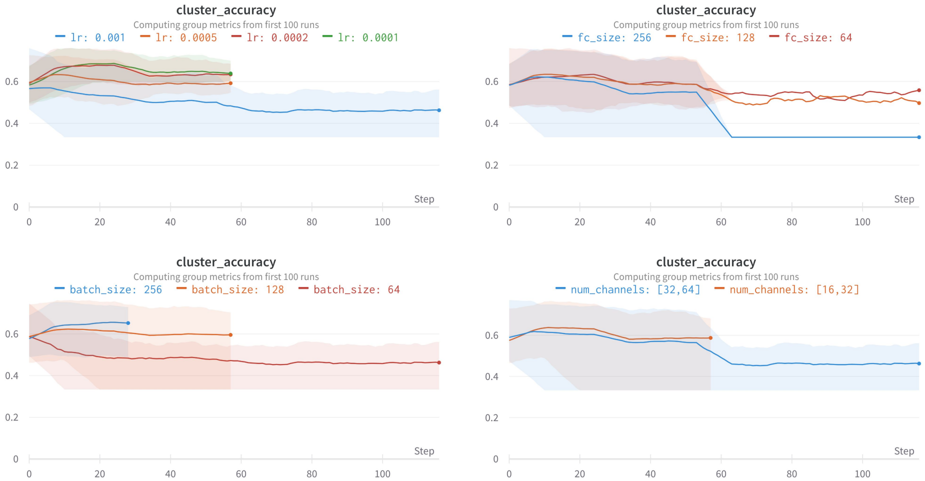
Output A = $\{\alpha\}_{q,i}$ assigns vector α to cluster C_i.

$\alpha^T K_h \alpha$ - Parzen window estimate of the CS Divergence
$m^T K_h m$ - Push assignments to corner of the simplex
sum(triu(AA^T)) - Favors orthogonal cluster assignments

Kampffmeyer et al. Neu. Networks 113 (2019)

Learning rate and batch size are important hyperparameters for DDC

Parameters scanned:
Batch size: 64, 128, 256. Learning rates: (10^{-4}, $2*10^{-4}$, $5*10^{-5}$, 10^{-3}), num_channels: ([16, 32], [32, 64]), num_depth/filter_size: ([5, [3,3]], [10, [5,5]]), final fully connected layer size: (64, 128, 256)



Datasets	Method	NMI	ACC[%]
MNIST	K-means	0.50	53.33
	ITC (parzen)	-	-
	ITC (kNN)	-	-
	SEC	0.77	68.82
	LDMGI	0.81	83.03
	DEC	0.81	84.31
	DDC	0.83	86.58
	DDC-VOTE	**0.87**	**88.49**
SEALS-3	K-means	0.13	51.33
	ITC (parzen)	0.003	35.30
	ITC (kNN)	0.10	53.95
	SEC	0.15	49.00
	LDMGI	0.13	50.43
	DEC	**0.17**	50.33
	DDC	0.14	**55.97**
	DDC-VOTE	0.13	53.30
SEALS-2	K-means	0.015	56.85
	ITC (parzen)	0.003	51.55
	ITC (kNN)	0.020	57.20
	SEC	0.021	58.15
	LDMGI	0.018	57.85
	DEC	0.005	54.04
	DDC	0.15	72.05
	DDC-VOTE	**0.18**	**74.65**

- Learning rates of 10^{-4} are optimal (ADAM optimizer)
- Batch sizes of 256 significantly better than 64.
- Cluster accuracy changes little when varying FC layer size / length of input sequence

Outline

1. Unsupervised Machine Learning
2. Dataset
3. Approach 1: Generative Adversarial Networks
4. Approach 2: Deep Divergence-based Clustering
5. Take-Away and Outlook

Take-away and outlook

- Unsupervised deep learning promises to tap into deep-learnings scaling with data while avoiding expensive data labelling
- Deep learning architectures can be trained to cluster ECE imaging data in an end-to-end fashion
- Methods typically rely on information-theoretic concepts, such as entropy and divergences. Unique requirements, such as prior on class distribution.
- Both, GAN and DDC achieve 60-70% clustering accuracy when tasked to cluster ECEI measurements of MHD phenomena
- Explore interpretability of cluster assignments
- Explore more recent developments in deep clustering algorithms

Code available online: https://github.com/rkube/ecei_generative