AutoBayes:
Automated Machine Learning with Bayesian Graph Exploration for Nuisance-Robust Inference

Toshiaki Koike-Akino

Andac Demir

Ye Wang

Deniz Erdogmus

Aug. 2020

MITSUBISHI ELECTRIC RESEARCH LABORATORIES (MERL)
Cambridge, Massachusetts, USA
http://www.merl.com

Thomas Bayes
1702 - 1761

Outline

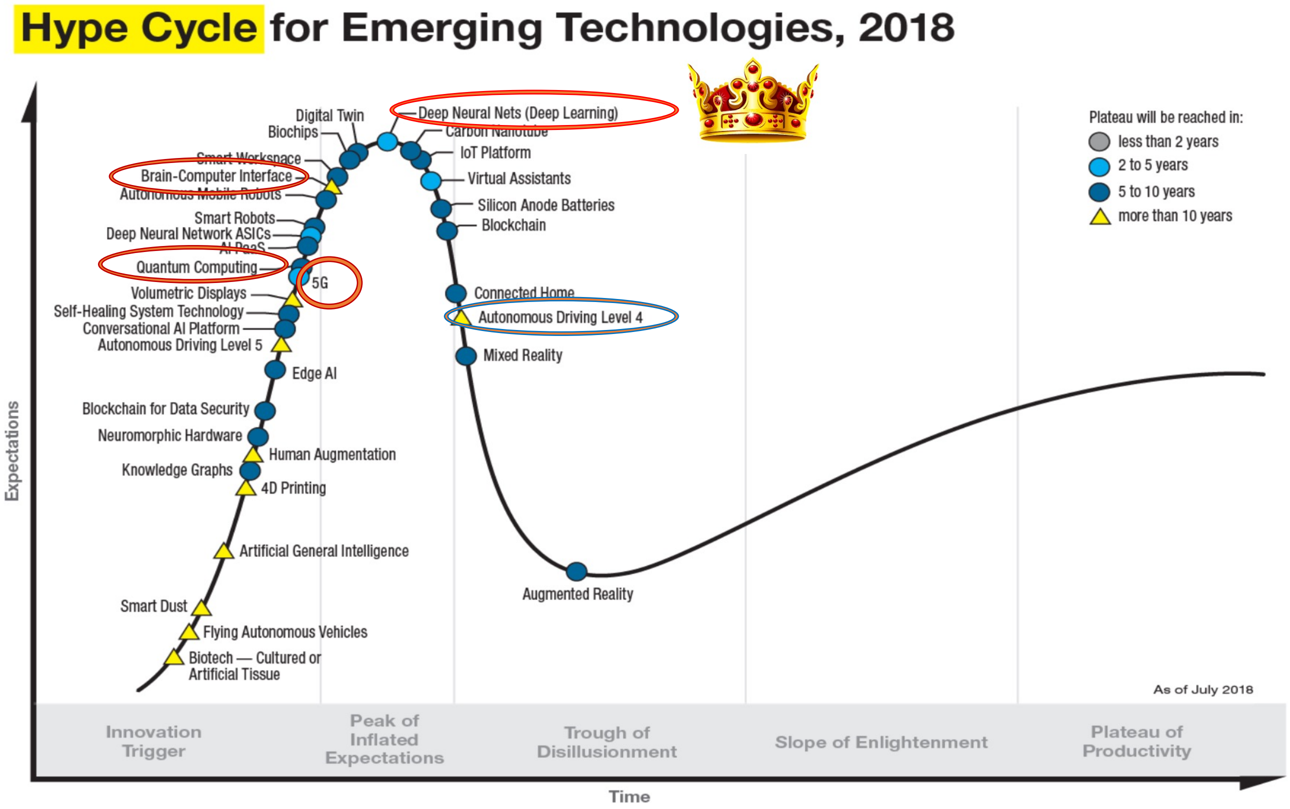
- Part I: Trends of machine learning
- Part II: **Adversarial learning** for nuisance-robust data analysis
- Part III: **Meta learning**: Automated machine learning (AutoML)
  - Automated architecture and hyperparameter tuning
- Part IV: AutoBayes
  - Bayesian inference graph modeling
  - Bayes Ball algorithm
- Part V: **Ensemble learning**
- Summary

Answering…
What is best DNN architecture?

Part I

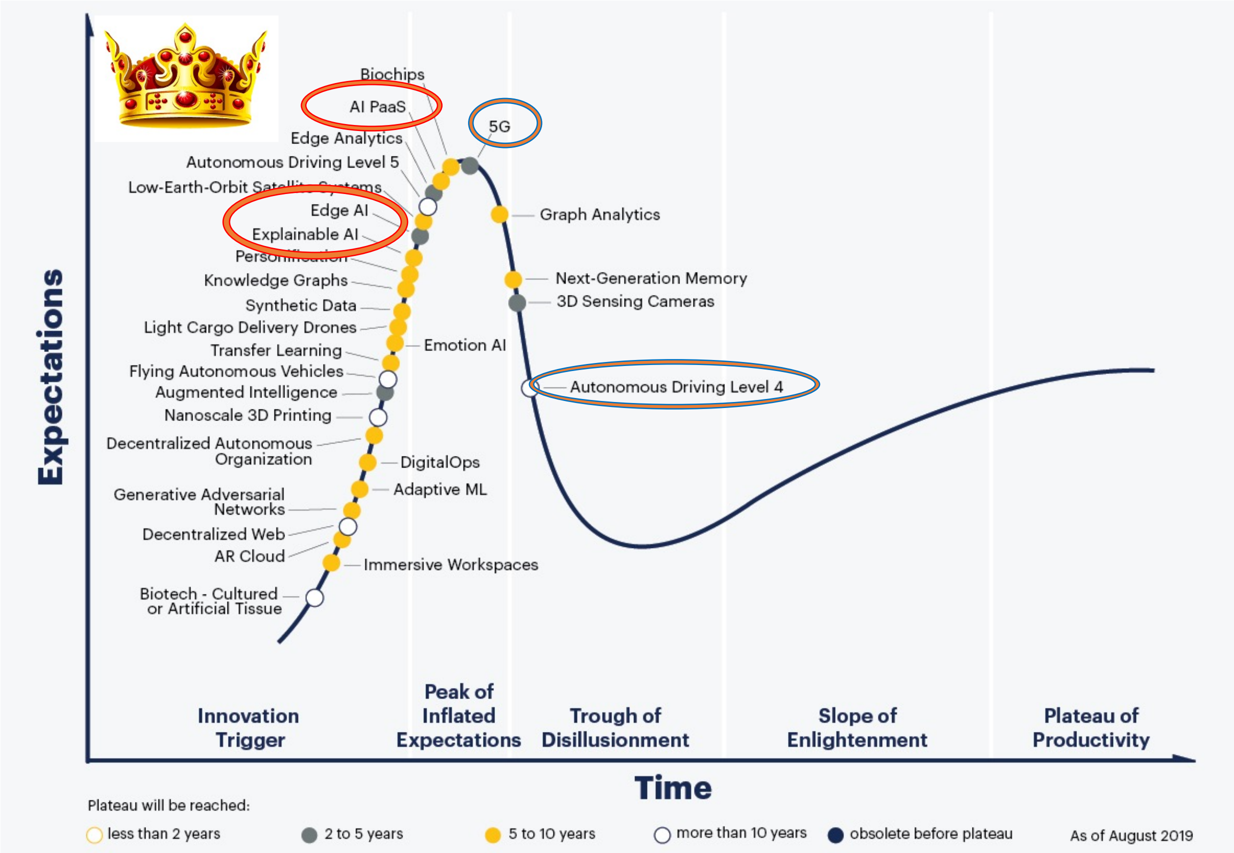
Emerging Technologies 2018

- Gartnar’s Hype Cycle for Emerging Technologies, 2018 July



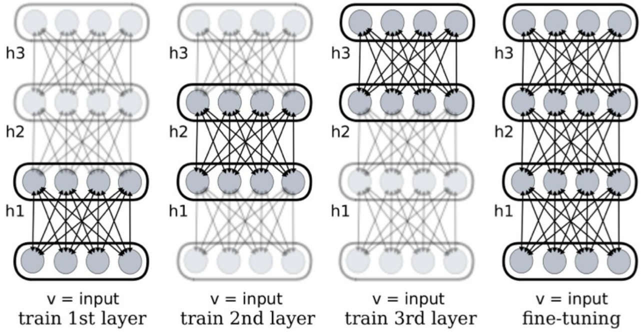
Emerging Technologies 2019 (Latest as of Aug. 2020)

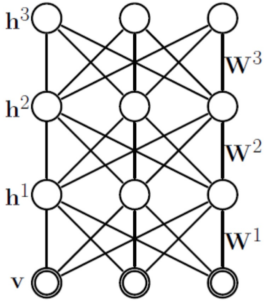
- Gartnar’s Hype Cycle for Emerging Technologies, 2019 August



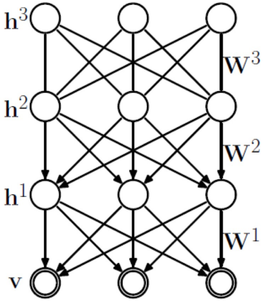
Deep Learning (DL) for Artificial Intelligence (AI)

- Deep learning = fancy name of multi-layer perceptron neural networks.
  - 2006 Hinton: Many layers, layer-wise pre-training, massive data sets
- Massively parallel computation
  - Driver: graphic processor units, tensor processor units …
- Variants:
  - Deep belief networks
  - Deep convolutional networks
  - Deep recurrent networks
  - Deep Boltzmann machines
  - Deep autoencoder

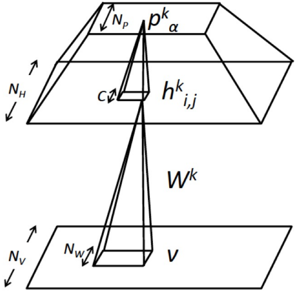




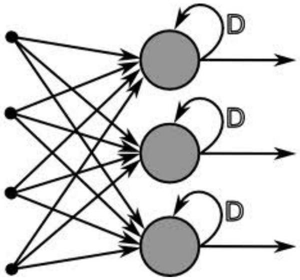
Deep Boltzmann Machine



Deep Belief Network



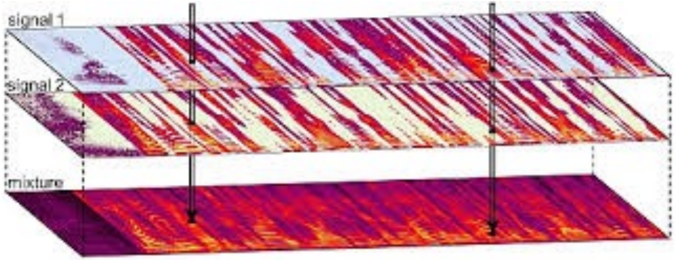
Convolutional Networks

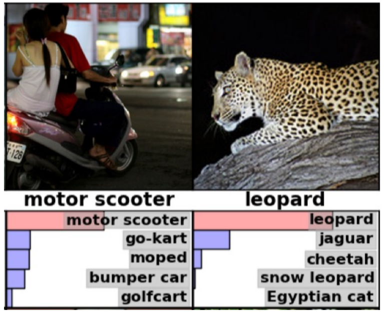


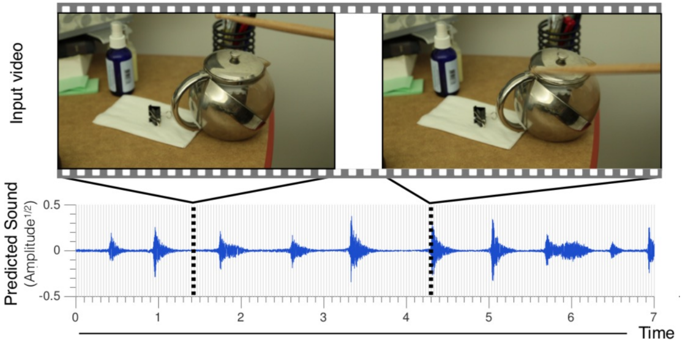
Recurrent Networks

Deep Learning for Media Signal Processing

- Audio & Visual Applications

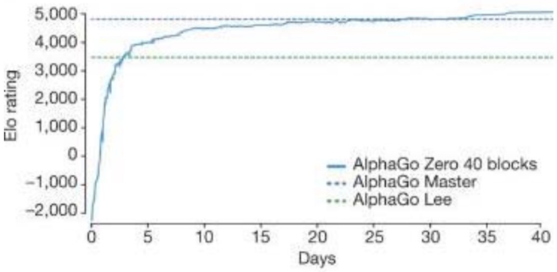


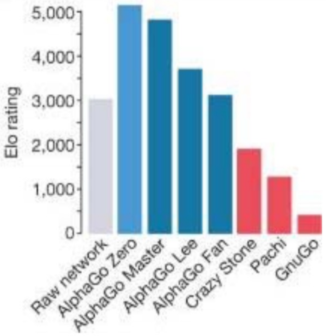




"man in black shirt is playing guitar."

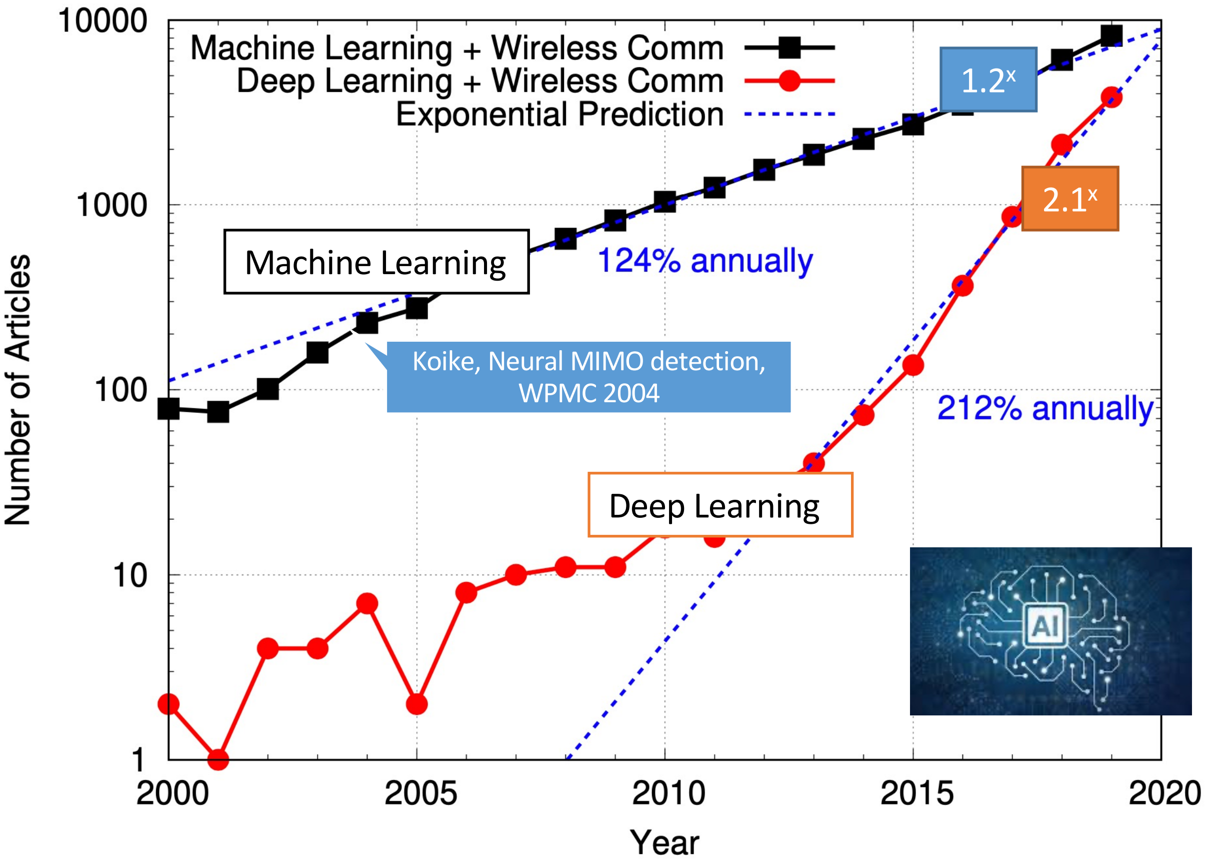
AI Surpassing Human-Level Performance

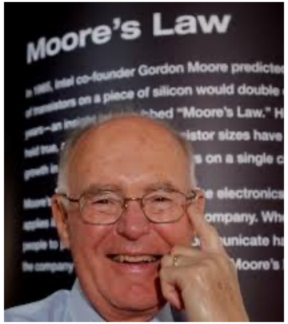


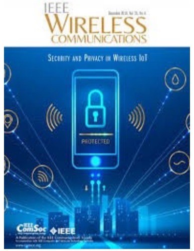


Moore’s Law: Exponential Growth in Applications

- Hit count of articles per year in GoogleScholar; ***Wireless Communication*** applications

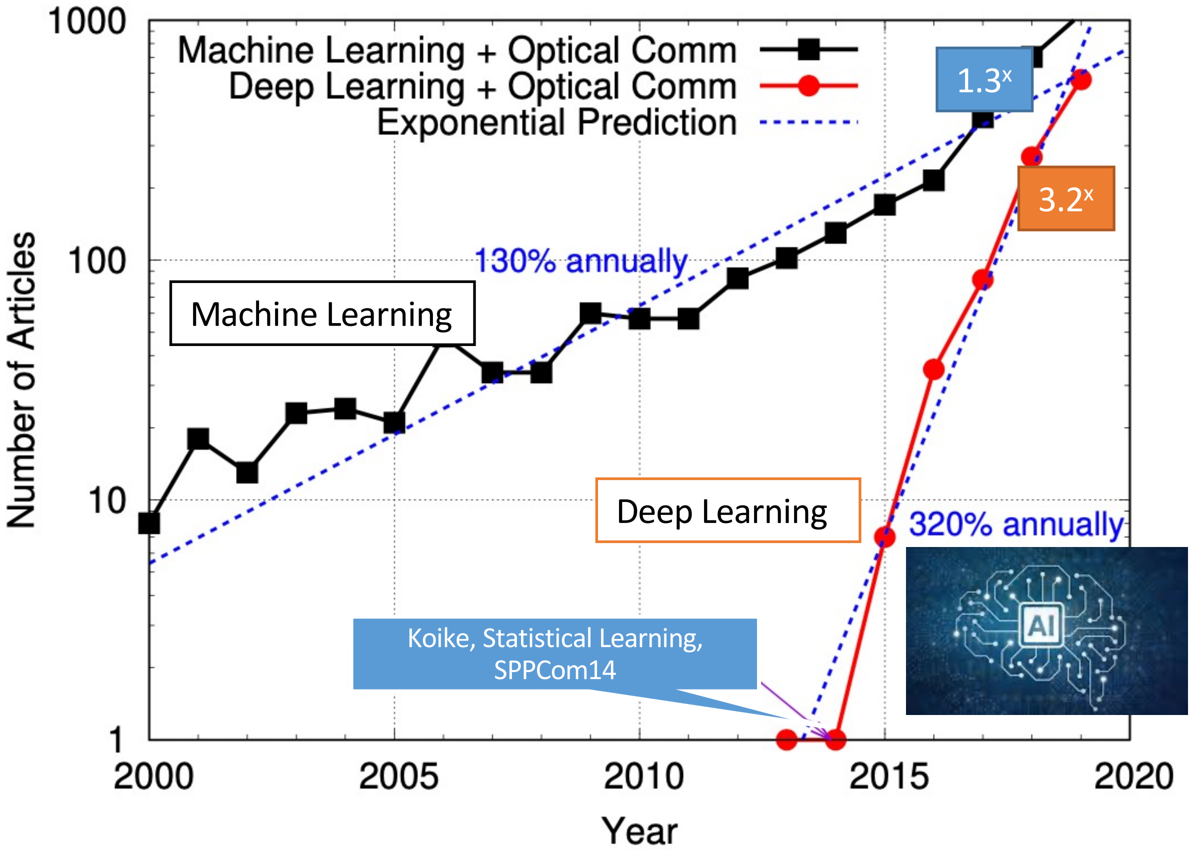


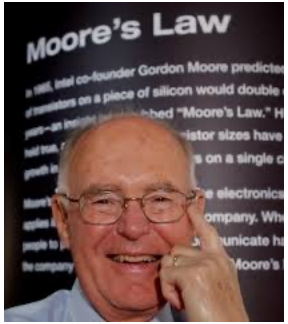


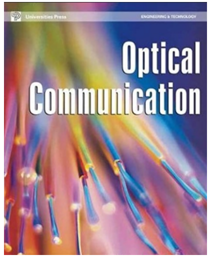


Moore's Law: Exponential Growth in Applications II

- Hit count of articles per year in GoogleScholar; ***Optical Communication*** applications







AI for ???

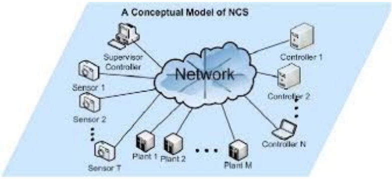
Applied Deep Learning

- AI has been applied to various fields

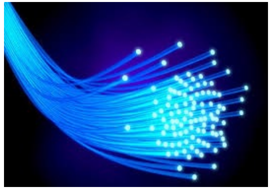
AI: Essential Component for R&D

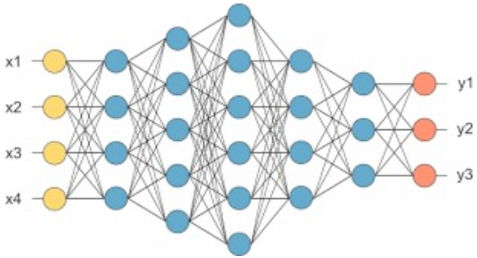
Wireless Communication

Networked Control



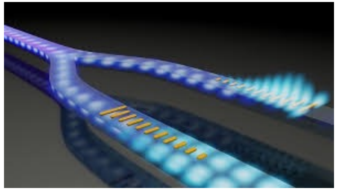
Optical Communication



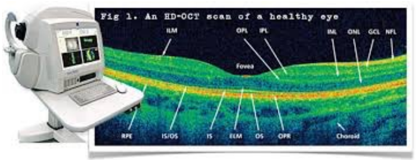


Localization Navigation

Device / Integrated Circuit



Tomography Imaging

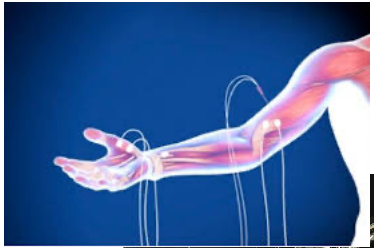


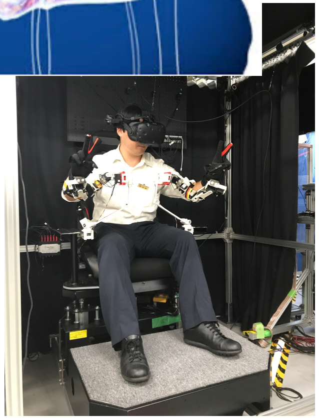
Bio-Sensing Human Interface

Biosignal Processing and Mind Sensing

- Joint work with Prof. Deniz Erdogmus (Northeastern Univ.)
- 2015 Ruhi Mahajan
  - Authentication (EMBC)
- 2016 Fernando Quivira
  - Probabilistic GMM+LSTM (BHI)
- 2017 Chun-Shu Wei
  - Few-shot learning (NER)
- 2018 Ozan Ozdenizci
  - Adversarial VAE (NER, SPL, Access)
- 2019 Mo Han
  - Complementary adversarial (EMBC, SPL)
  - Rateless soft disentangling (JBHI)
- 2020 Andac Demir
  - AutoBayes (Access)
  - Graph EEG net (EMBC)

Northeastern Univ.

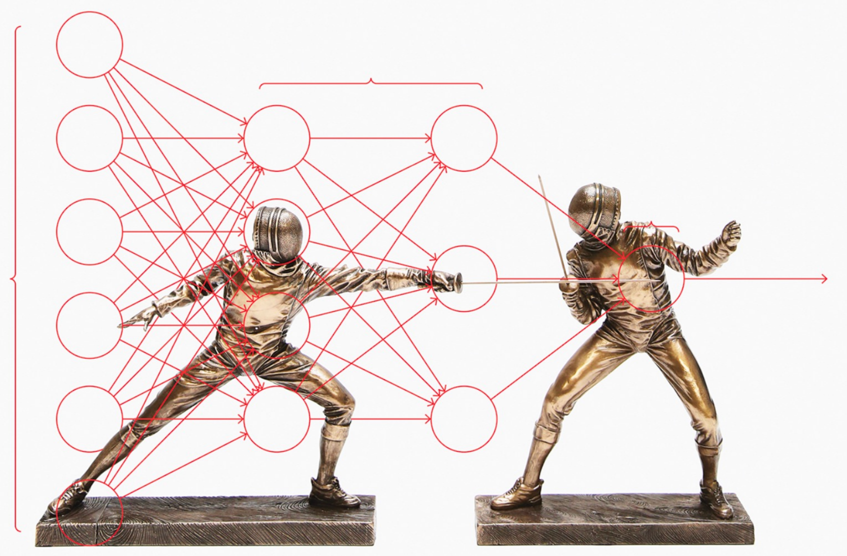




Our Publications

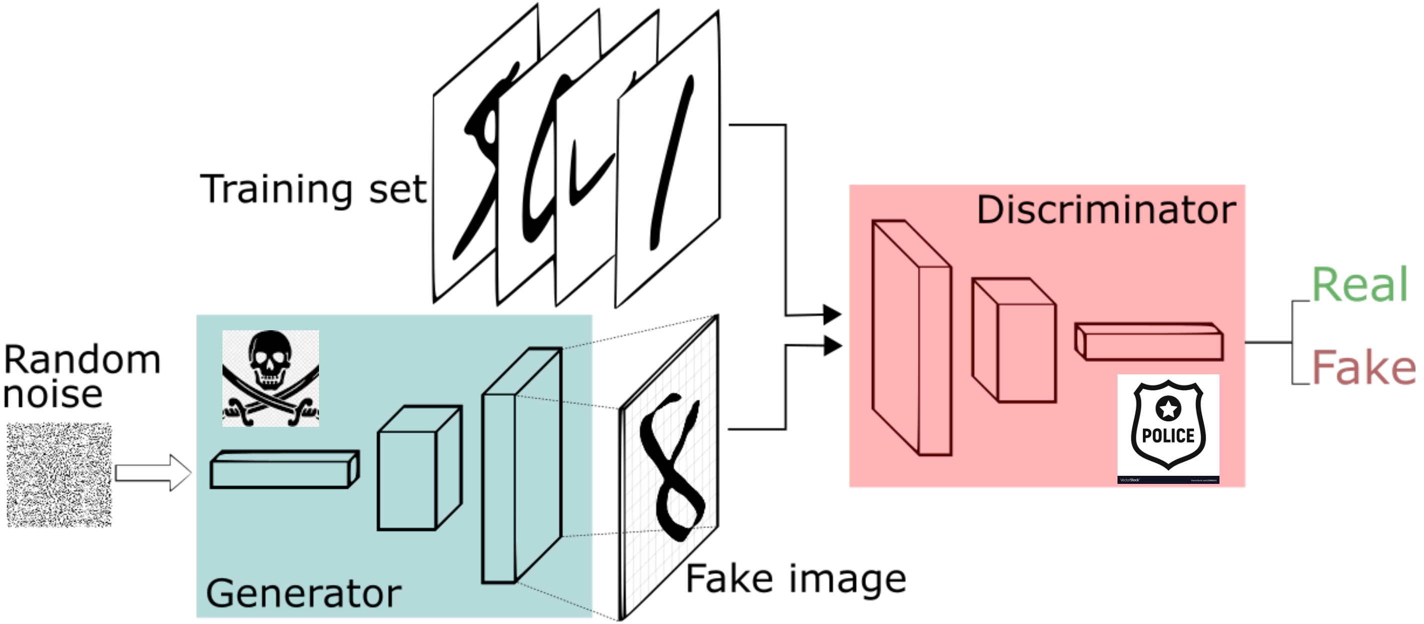
1. Koike-Akino, T., Mahajan, R., Marks, T.K., Tuzel, C.O., Wang, Y., Watanabe, S., Orlik, P.V., **"High-Accuracy User Identification Using EEG Biometrics"**, IEEE EMBC, August 2016.
2. Wang, Y., Koike-Akino, T., Erdogmus, D. “**Invariant Representations from Adversarially Censored Autoencoders**”, arxiv:1805.08097, May 2018.
3. Quivira, F., Koike-Akino, T., Wang, Y., Erdogmus, D., **"Translating sEMG Signals to Continuous Hand Poses using Recurrent Neural Networks"**, IEEE BHI, January 2018.
4. Wei, C.-S., Koike-Akino, T., Wang, Y., **"Spatial Component-wise Convolutional Network (SCCNet) for Motor-Imagery EEG Classification"**, IEEE NER, March 2019.
5. Ozdenizci, O., Wang, Y., Koike-Akino, T., Erdogmus, D., **"Transfer Learning in Brain-Computer Interfaces with Adversarial Variational Autoencoders”**, IEEE NER, March 2019. (arxiv:1812.06857, Dec. 2018)
6. Ozdenizci, O., Wang, Y., Koike-Akino, T., Erdogmus, D., **"Adversarial Deep Learning in EEG Biometrics"**, IEEE SPL, March 2019. (arxiv:1903.11673, May 2019)
7. Ozdenizci, O., Wang, Y., Koike-Akino, T., Erdogmus, D., **"Learning Invariant Representations from EEG via Adversarial Inference"**, IEEE Access , April 2020.
8. Koike-Akino, T., Wang, Y., **"Stochastic Bottleneck: Rateless Auto-Encoder for Flexible Dimensionality Reduction"**, IEEE ISIT, June 2020. (arxiv:2005.02870, May 2020)
9. Han, M., Ozdenizci, O., Wang, Y., Koike-Akino, T., Erdogmus, D., **"Disentangled Adversarial Transfer Learning for Physiological Biosignals"**, IEEE EMBC, July 2020. (arxiv:2004.08289, Apr. 2020)
10. Han, M., Ozdenizci, O., Wang, Y., Koike-Akino, T., Erdogmus, D., **" Disentangled Adversarial Autoencoder for Subject-Invariant Physiological Feature Extraction "**, IEEE SPL, July 2020. (arxiv:2008.11426)
11. Demir, A., Koike-Akino, T., Wang, Y., Erdogmus, D., **"AutoBayes: Automated Bayesian Graph Exploration for Nuisance-Robust Inference"**, IEEE Access, Mar. 2021. (*arxiv:2007.01255*, July 2020).
12. Haruna, M., Ogino, M., Koike-Akino, T., “**Proposal and Evaluation of Visual Haptics for Manipulation of Remote Machine System**,” Frontiers, Aug. 2020.
13. Han, M., Ozdenizci, O., Wang, Y., Koike-Akino, T., Erdogmus, D., **" Universal Physiological Representation Learning with Soft-Disentangled Rateless Autoencoders"**, IEEE JBHI, Mar. 2021. arxiv:2009.13453

Adversarial Learning



Adversarial Networks

- Generative Adversarial Networks (GAN) [Goodfellow et al, 2014]
  - Train two **competing** neural networks
  - Generator learns to fake images by trying to fool Discriminator



- Competition between counterfeiters and police ⇒ better fake money

GAN for Synthetic Faces

- Nvidia GAN Results [Karras et al, 2018]

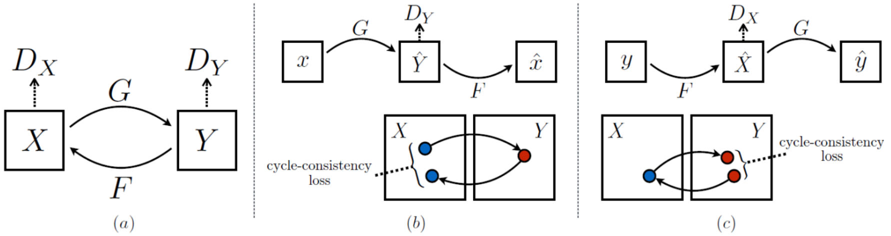
Realistic Fake Faces

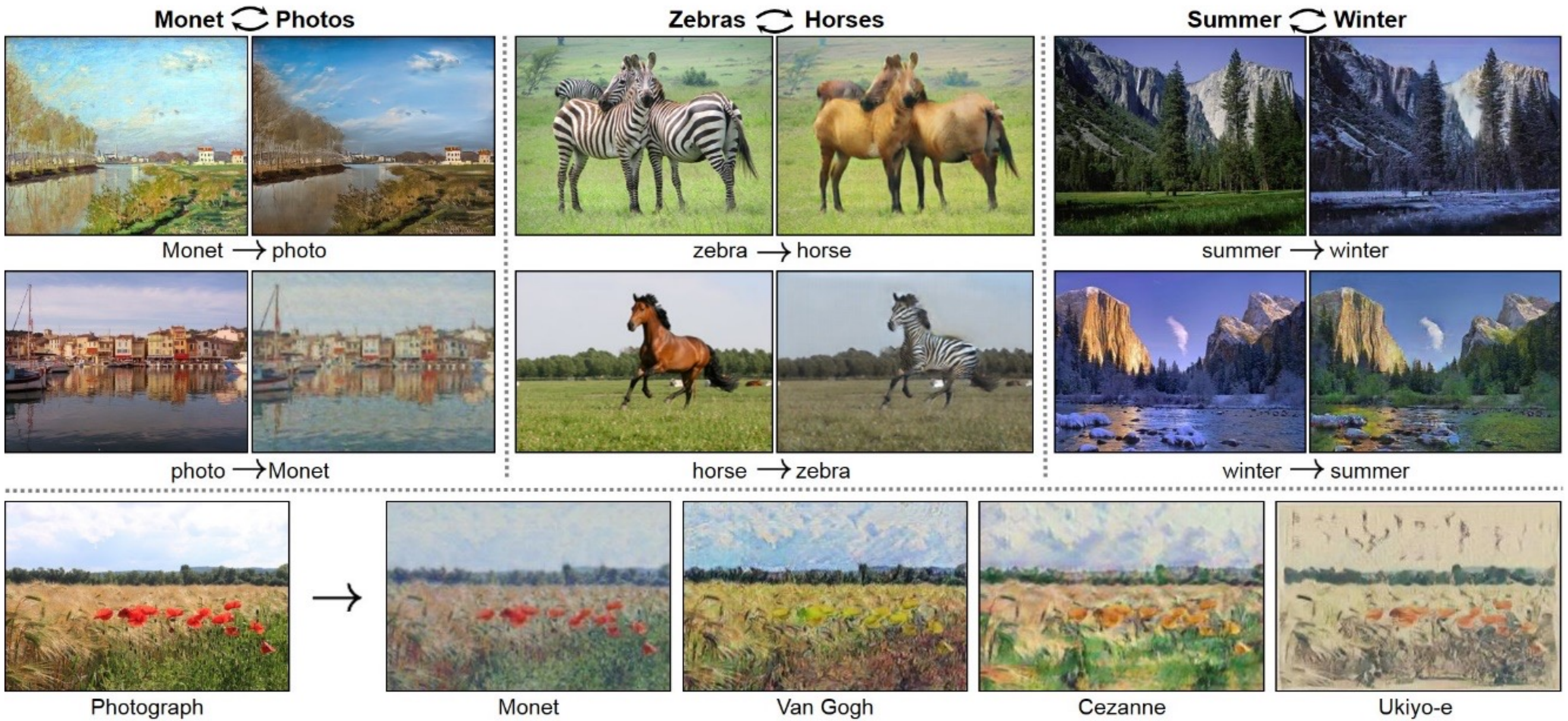
youtube:XOxxPcy5Gr4

youtube:kSLJriaOumA

CycleGAN for Image Translation

- CycleGAN [Zhu et al, 2017]

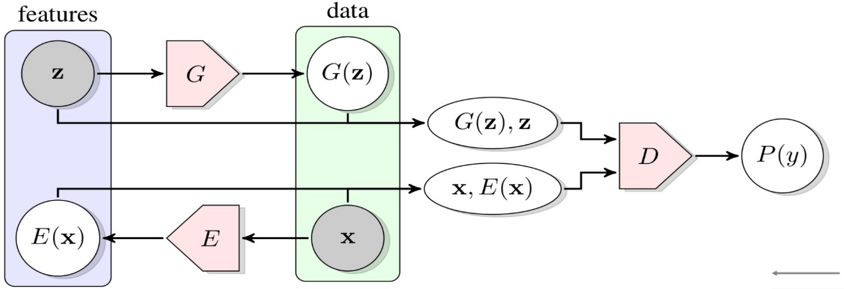


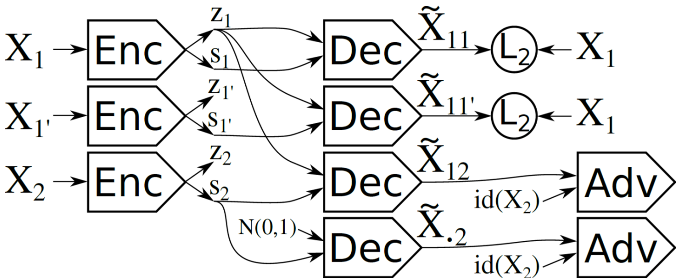


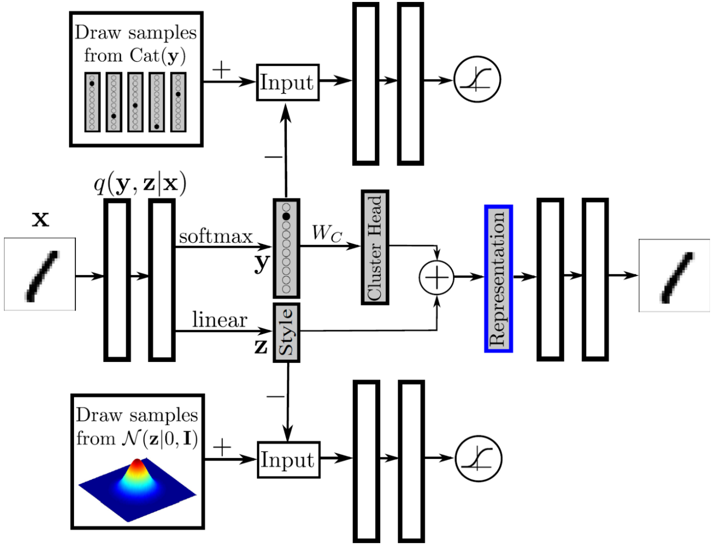
Style Change

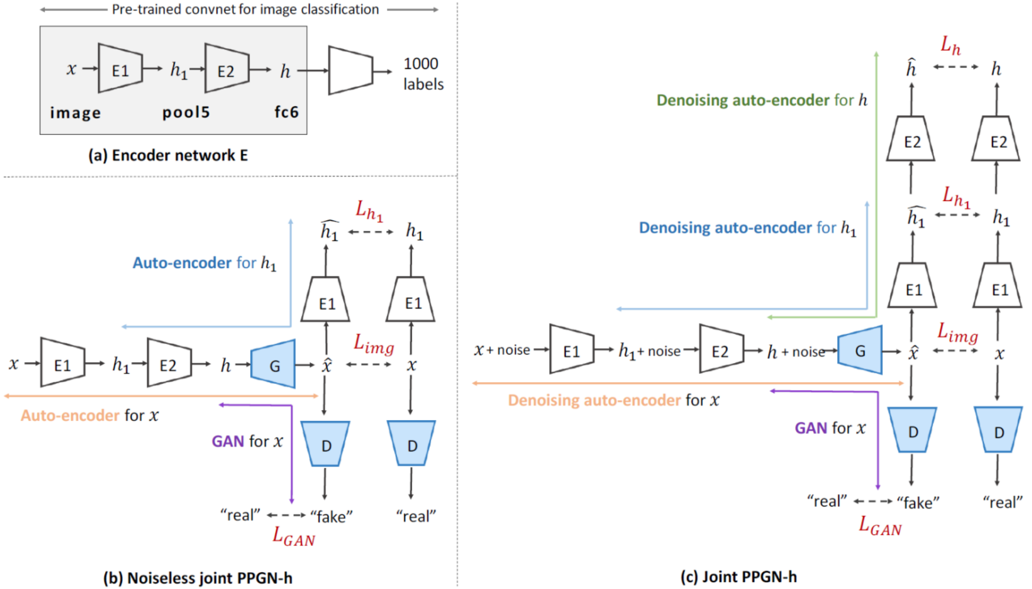
GAN Zoo – Lots of Adversarial Networks

- Many different ways to adversarially combine networks

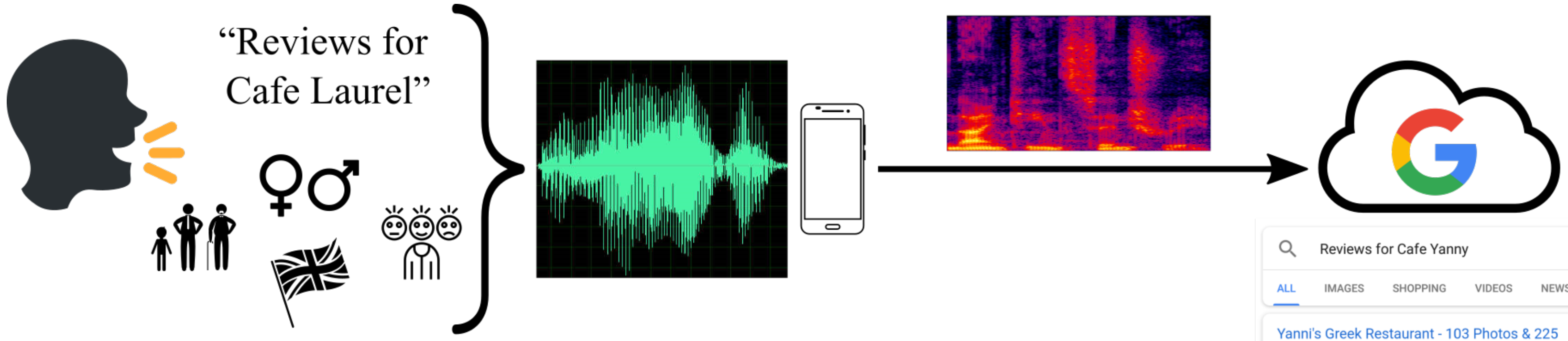


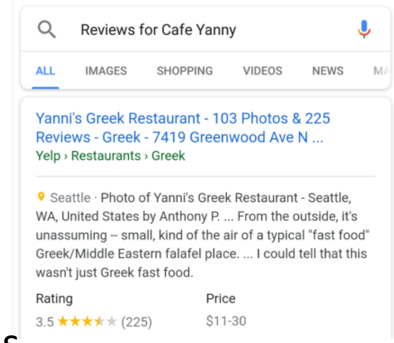






Learning Nuisance-Invariant Data Representations

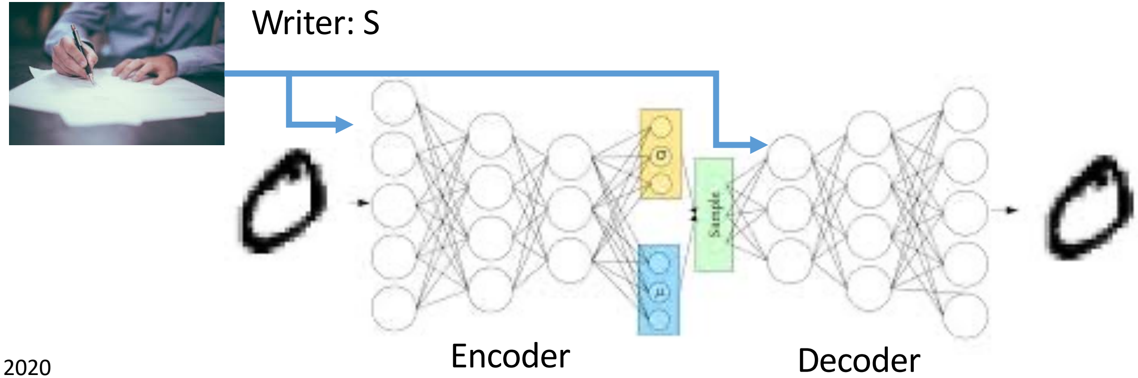




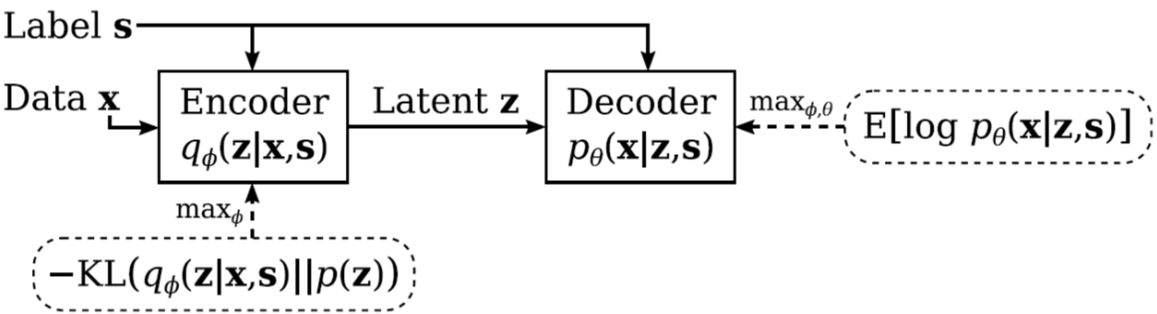
- Objective: extract invariant representations (features)
  - Remove nuisance variations, sensitive attributes
  - Motivation: transferability, generalizability, robustness, privacy, fairness
- Autoencoder model: data $\mathbf{x} \rightarrow \boxed{\text{Enc}} \rightarrow$ latent $\mathbf{z} \rightarrow \boxed{\text{Dec}} \rightarrow \hat{\mathbf{x}}$
  - General purpose data representations $\mathbf{z}$
  - Can also support translation, feature/style transfer

Variational AutoEncoders (VAE)

- VAE introduced by [Kingma & Welling, 2014] with conditional extension [Sohn et al, 2015]
- Learn CVAE model: $(\mathbf{x}, \mathbf{s}, \mathbf{z}) \sim p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{s})p(\mathbf{s})p(\mathbf{z})$
  - $\mathbf{x}$ raw data features
  - $\mathbf{s}$ nuisance variations (conditioning variable)
  - $\mathbf{z}$ latent (unobserved) representation
  - **Invariance:** model explicitly specifies independence between $\mathbf{s}$ and $\mathbf{z}$
  - Generative model $p(\mathbf{x}|\mathbf{z}, \mathbf{s})$ from appropriate parametric family
  - Convenient latent model $p(\mathbf{z}) = N(\mathbf{0};\ \mathbf{I})$
  - Nuisance model $p(\mathbf{s})$ arbitrary (not used for training)



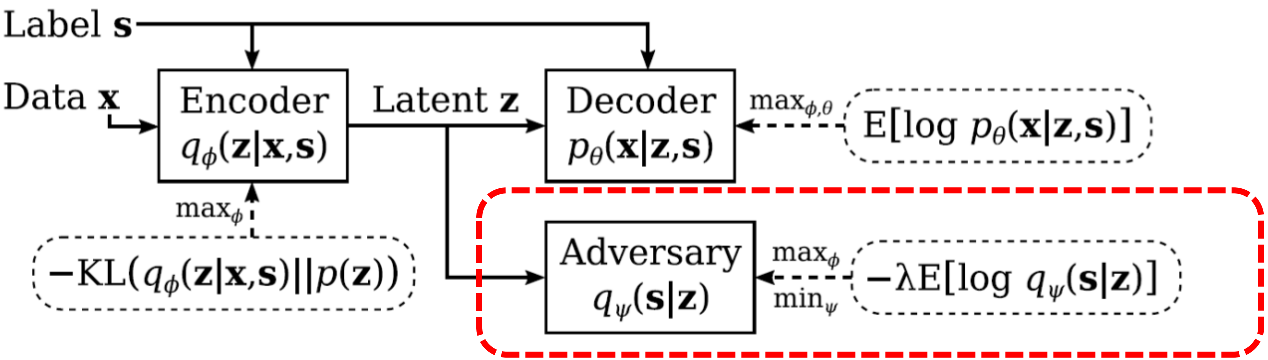
VAE Training



- Decoder: generative model $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{s})$
- Encoder: variational posterior $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{s})$
  - In principle, $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{s}) \to p_\theta(\mathbf{z}|\mathbf{x},\mathbf{s})$ and hence $\mathbf{z} \perp\!\!\!\perp \mathbf{s}$
- However, in practice, invariance ($I(\mathbf{s};\mathbf{z}) = 0$) needs to be enforced

$$\max_{\theta,\phi} \mathcal{L}(\theta,\phi) - \lambda I(\mathbf{s};\mathbf{z})$$

VAE Training with Adversarial Censoring



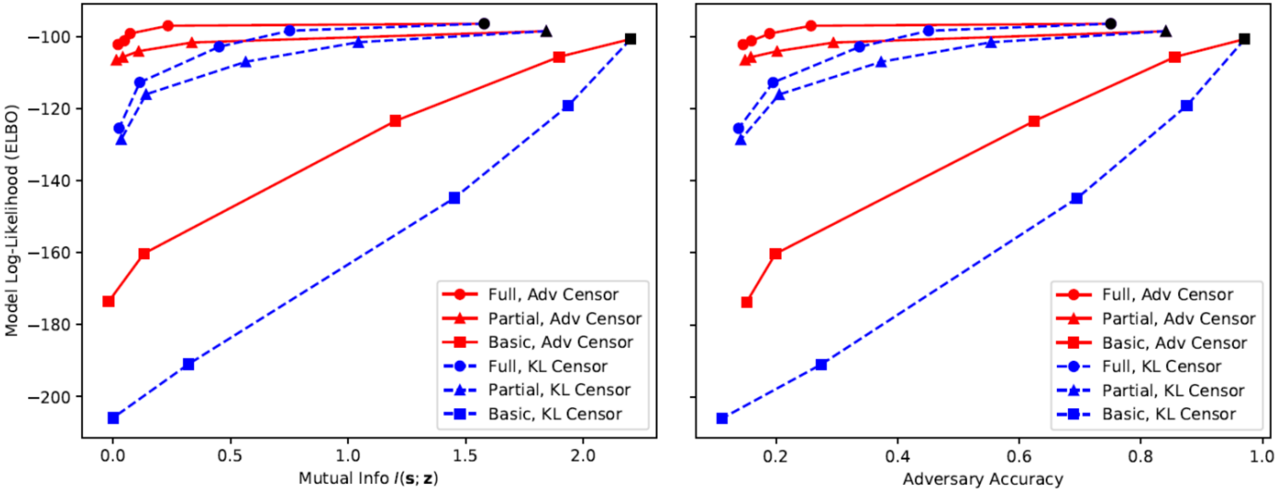
- Decoder: generative model $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{s})$
- Encoder: variational posterior $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{s})$
  - In principle, $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{s}) \to p_\theta(\mathbf{z}|\mathbf{x},\mathbf{s})$ and hence $\mathbf{z} \perp\!\!\!\perp \mathbf{s}$
- However, in practice, invariance $(I(\mathbf{s};\mathbf{z}) = 0)$ needs to be enforced

$$\max_{\theta,\phi} \mathcal{L}(\theta,\phi) - \lambda I(\mathbf{s};\mathbf{z}) \Rightarrow \max_{\theta,\phi} \min_{\psi} \mathcal{L}(\theta,\phi) \underbrace{-\lambda\mathbb{E}\left[\log q_\psi(\mathbf{s}|\mathbf{z})\right]}_{\geq -\lambda(I(\mathbf{z};\mathbf{s}) - h(\mathbf{s}))}$$

- Adversary $q_\psi(\mathbf{s}|\mathbf{z})$ attempts to recover $\mathbf{s}$, approximates $I(\mathbf{s};\mathbf{z})$

A-CVAE Benefit for MNIST Dataset

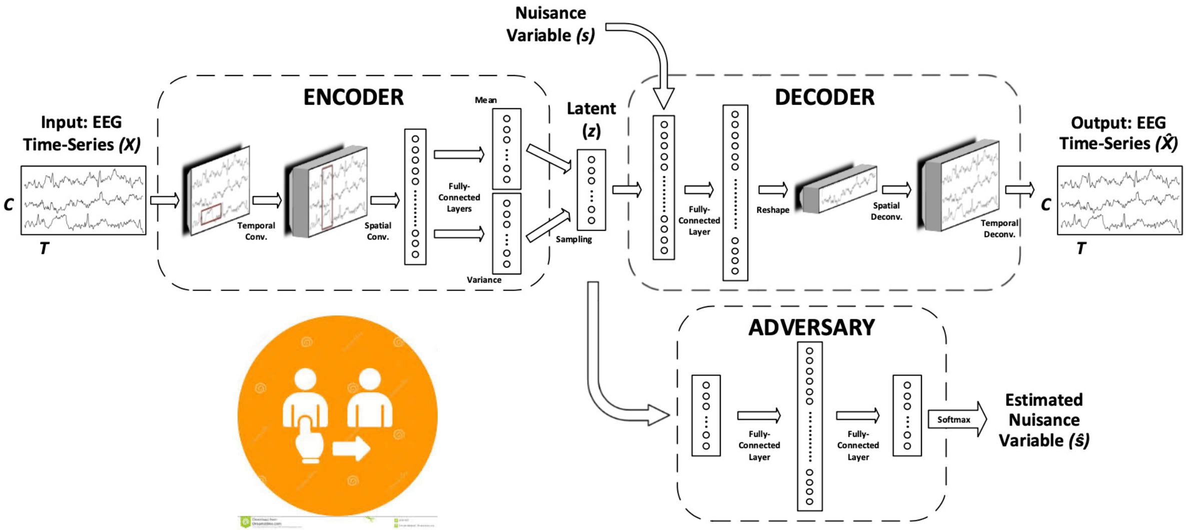
- Wang, Y., Koike-Akino, T., Erdogmus, D. "**Invariant Representations from Adversarially Censored Autoencoders**", arxiv:1805.08097, May 2018.

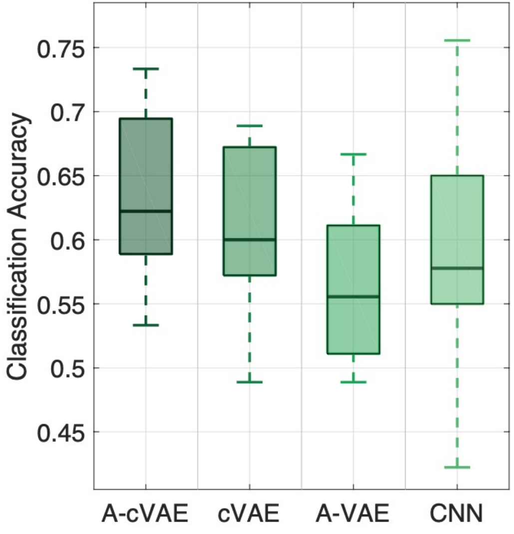


- Full (●): full VAE conditioned on s, i.e. $E(\mathbf{x},\mathbf{s})$, $D(\mathbf{z},\mathbf{s})$
- Partial (▲): only decoder conditioned on s, i.e. $E(\mathbf{x})$, $D(\mathbf{z},\mathbf{s})$
- Basic (■): no conditioning on s, i.e. $E(\mathbf{x})$, $D(\mathbf{z})$
- KL censoring alternative: use $-\gamma \mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{s}) \| p(\mathbf{z})\big)$, with $\gamma > 1$

A-CVAE for Nuisance-Robust Transfer Learning: Zero-Shot Learning

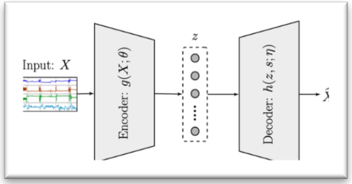
- Cross-subject transfer learning for BCI [Ozdenizci et al, NER'19]
  - Task: motor-imagery decoding from EEG measurements
  - Subject variability is the nuisance variation suppressed
- Cross-session EEG-based biometrics [Ozdenizci et al, SPL'19]
  - Task: subject identification from EEG measurements
  - Session variability is the nuisance variation suppressed

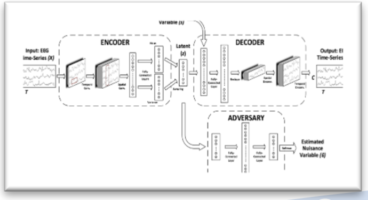


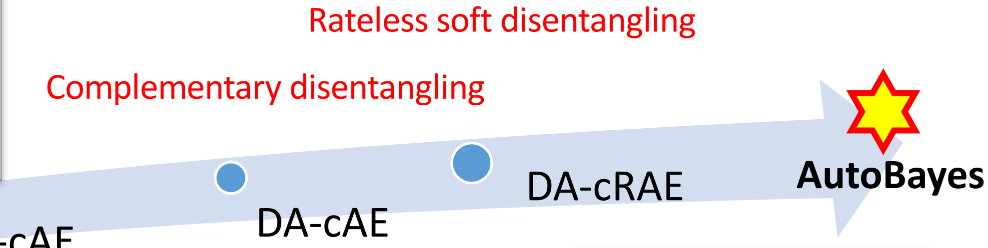


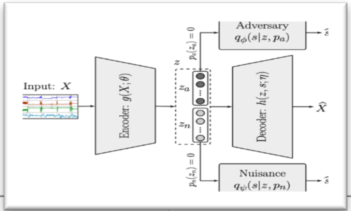
Subject transfer accuracy

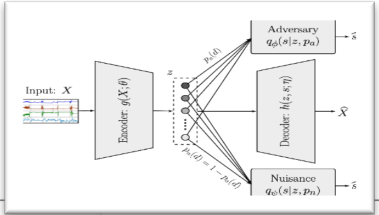
Evolution Map: Nuisance-Invariant Feature Extraction

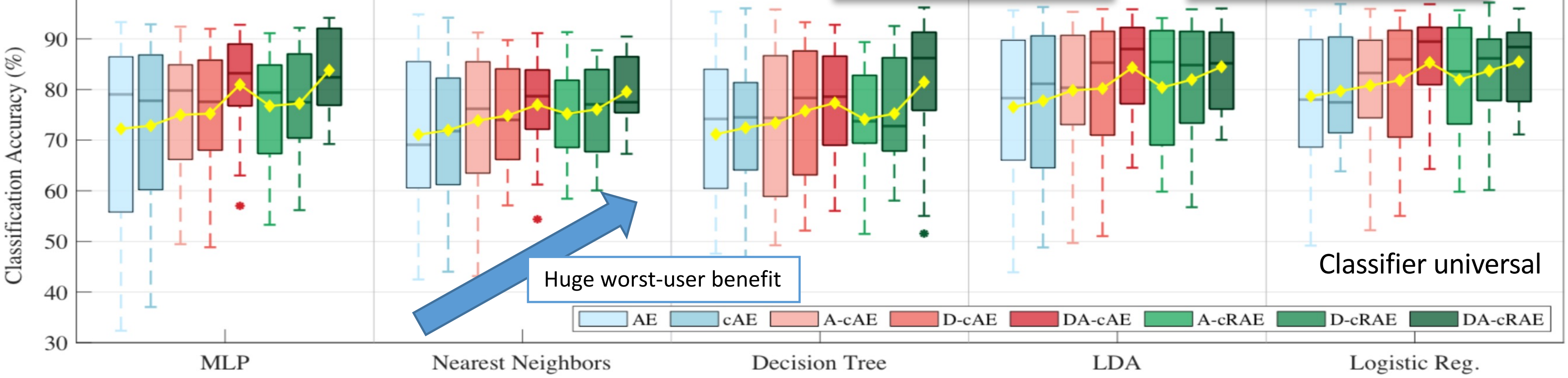








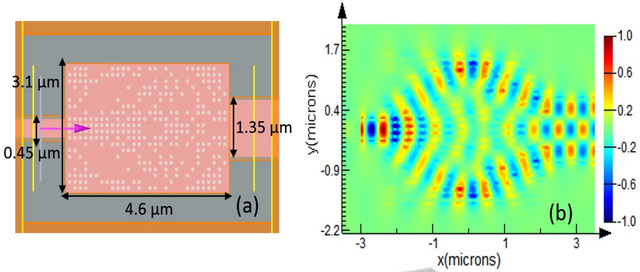


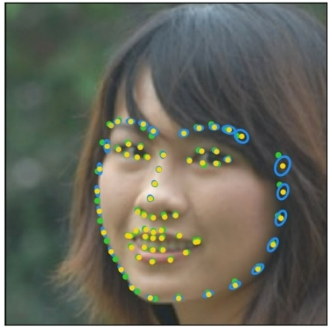


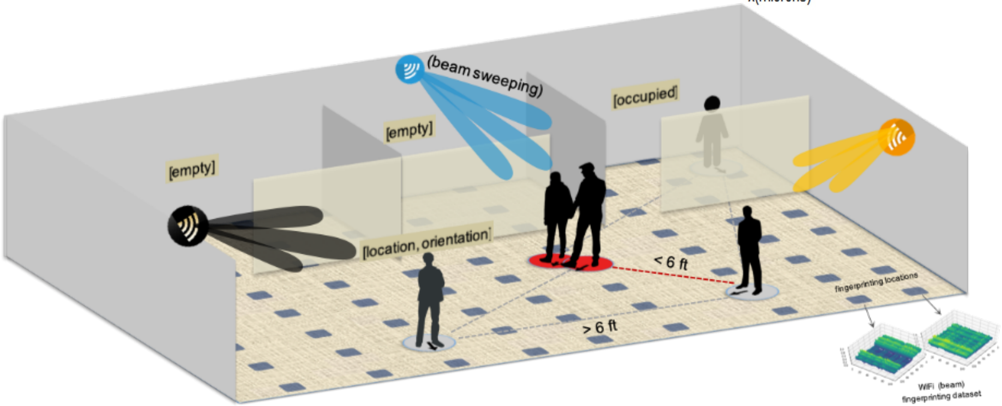
Subject transfer accuracy: Stress dataset (heartrate, temperature, etc.)

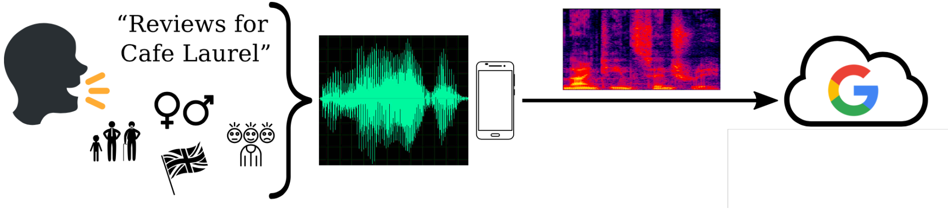
Nuisance-Robust Analysis: Not Only for Biosignal Applications

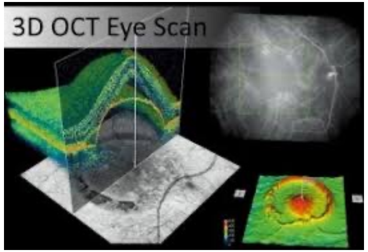
- Nano-photonic device design
- Privacy preserving
- Localization
- Image sensing
- Speech recognition
- Face recognition
- ...









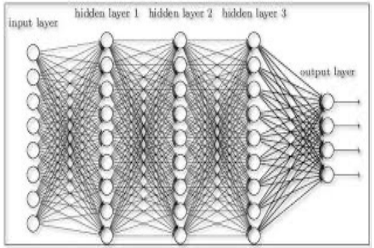


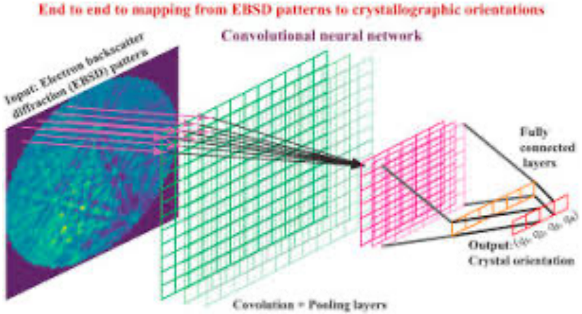
Part III

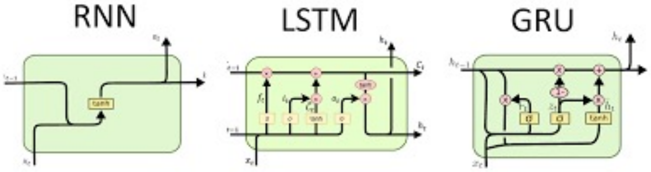
AutoML

Various DNN Architectures

- Forward, recursive, convolutional
- LSTM, GRU, Transformer
- Bottleneck, U-net, HR-net
- ResNet, loopy, clique
- Inception, bilinear







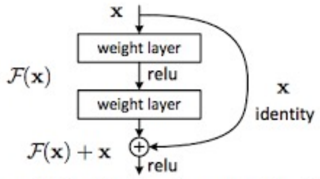
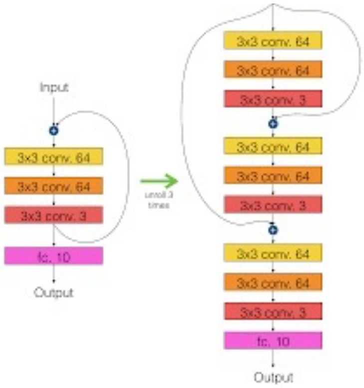


Figure 2. Residual learning: a building block.



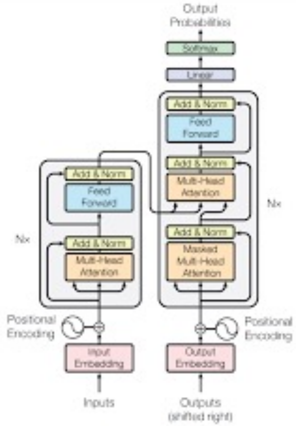
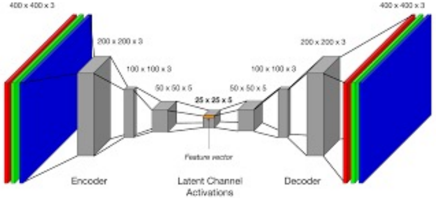
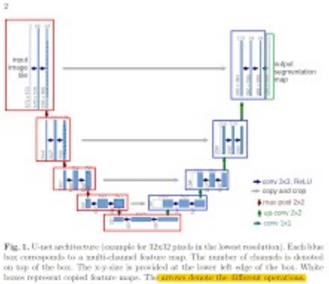
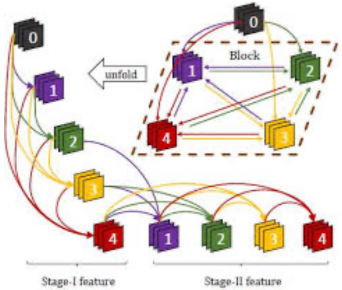
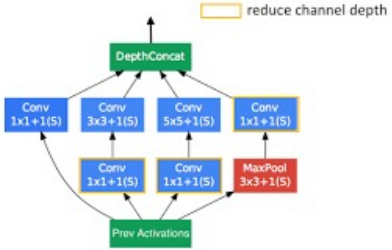


Figure 1: The Transformer - model architecture.





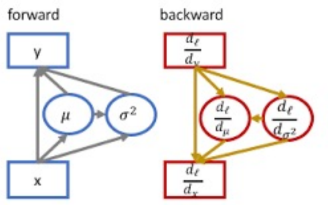




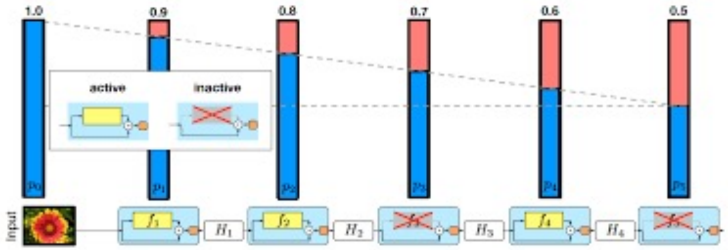
Various Hyperparameters

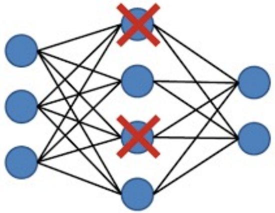
"***Undergraduate-student gradient***" trial-and-error

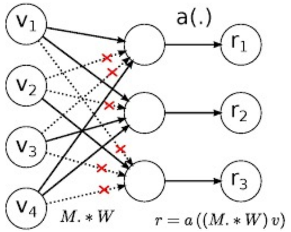
- Loss functions: CrossEntropy, MSE, MAE, hinge, L1, …
- Updaters: SGD, Adam, AdaDelta, AdaGrad, AdaMax, LBFGS, RMSprop, …
- Learning rate schedulers: CosineAnnealing, Cyclic, Exponential, MultiStep, OnPlateau, …
- Regularizations: Dropout, Batch Norm, Spectral Norm, DropConnect, StochasticDepth, ShakeDrop, Shake-Shake, …
- Activations: ReLU, sigmoid, tanh, …
- Augmentation, depth, width
- Quantization, initialization
- Pooling, unpooling, padding, …

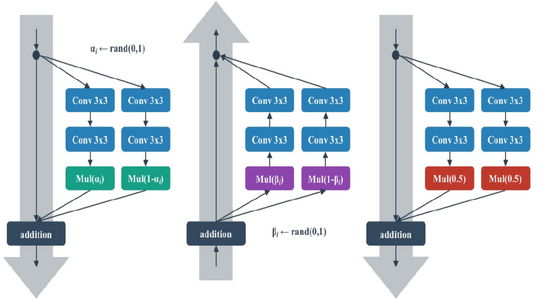


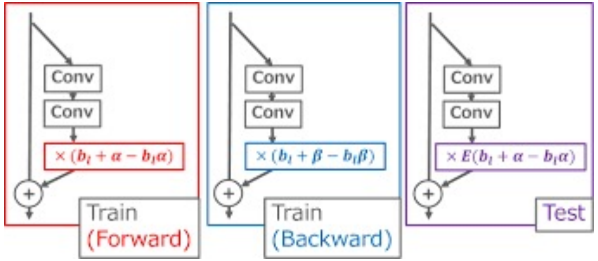
Batch Normalization [5] in training mode.







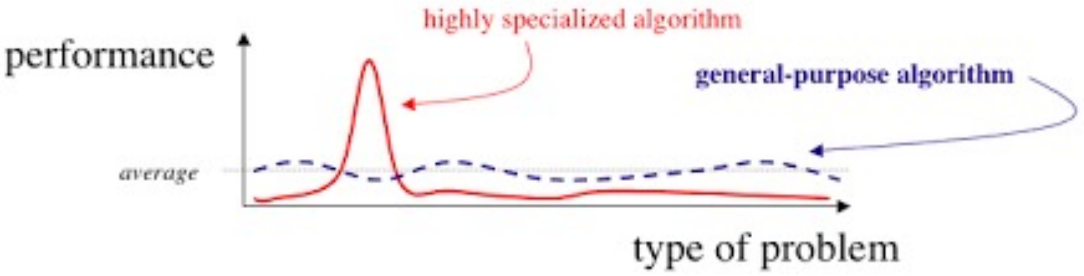


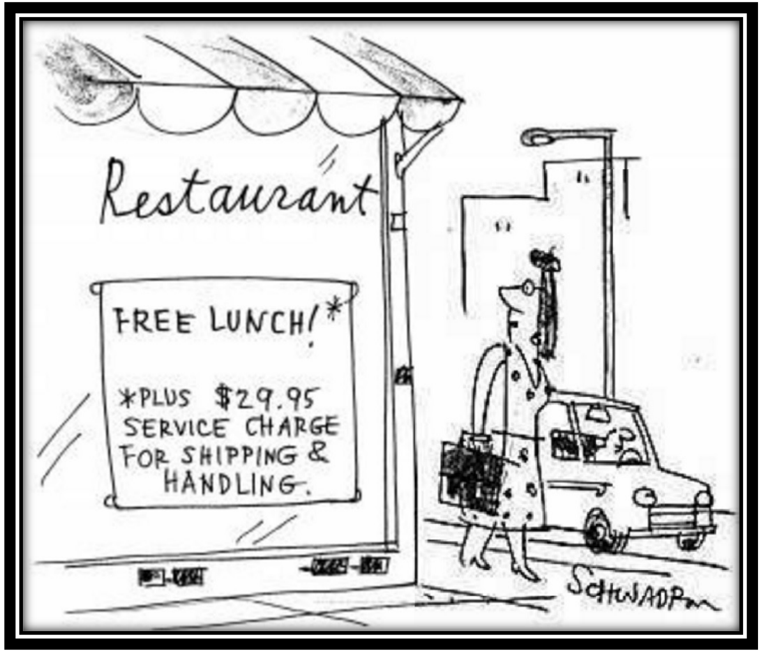


No Free Lunch Theorem

- Wolpert 1996: *What Does Dinner Cost?*, NASA Ames Research Center

- There is no one model that works **best for every problem**
- We thus shall **try multiple models** and find one that works **best for a particular problem**



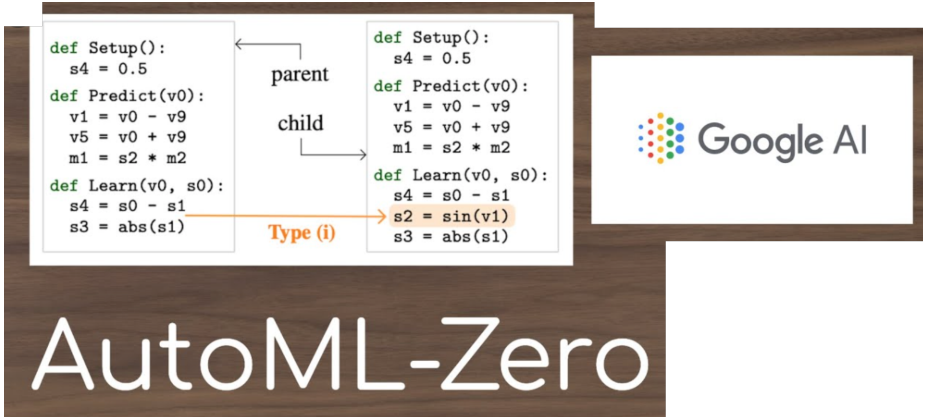


Automated Machine Learning: AutoML-Zero

Human Experts
Programming

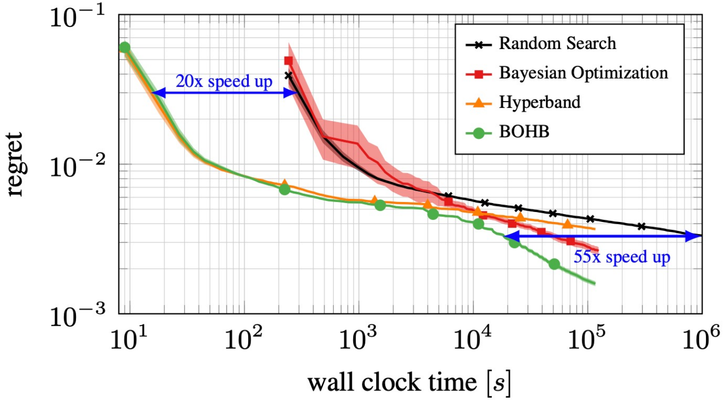
AI Experts
Evolutionary Programming

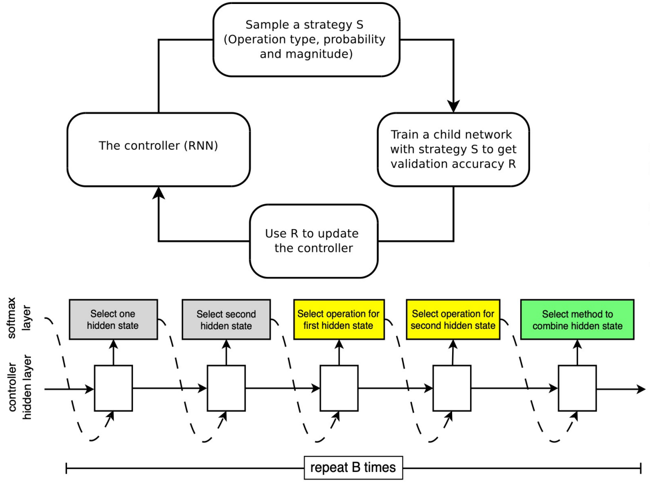
```
model = nn.Sequential(
            nn.Conv2d(1,20,5),
            nn.ReLU(),
            nn.Conv2d(20,64,5),
            nn.ReLU()
        )
```

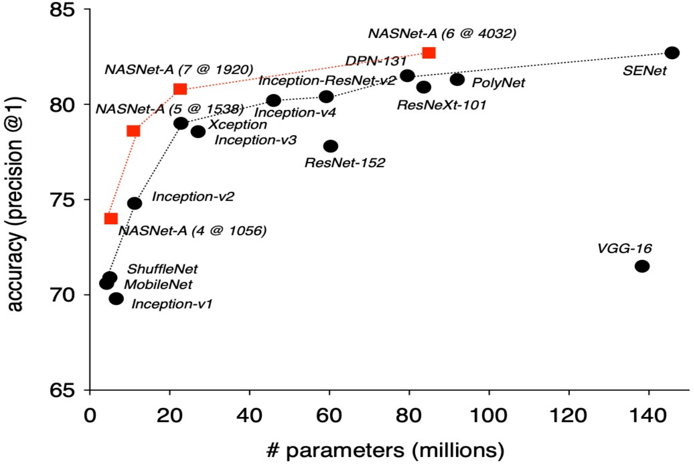


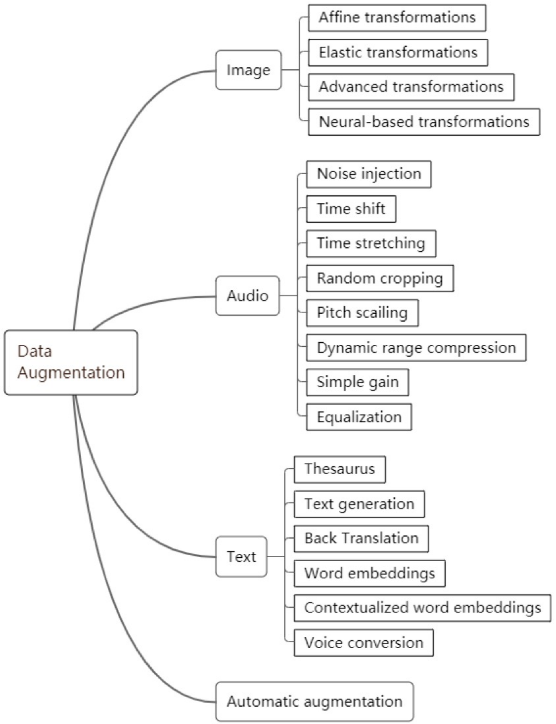
AutoML, Learning to Learn (L2L), Meta Learning

- Hyperparameter exploration: Auto-Pytorch, …
  - Bayesian optimization
  - Evolutionary optimizatior
- Architecture exploration
  - Reinforcement learning
  - Cell-based building
- Augmentation exploration



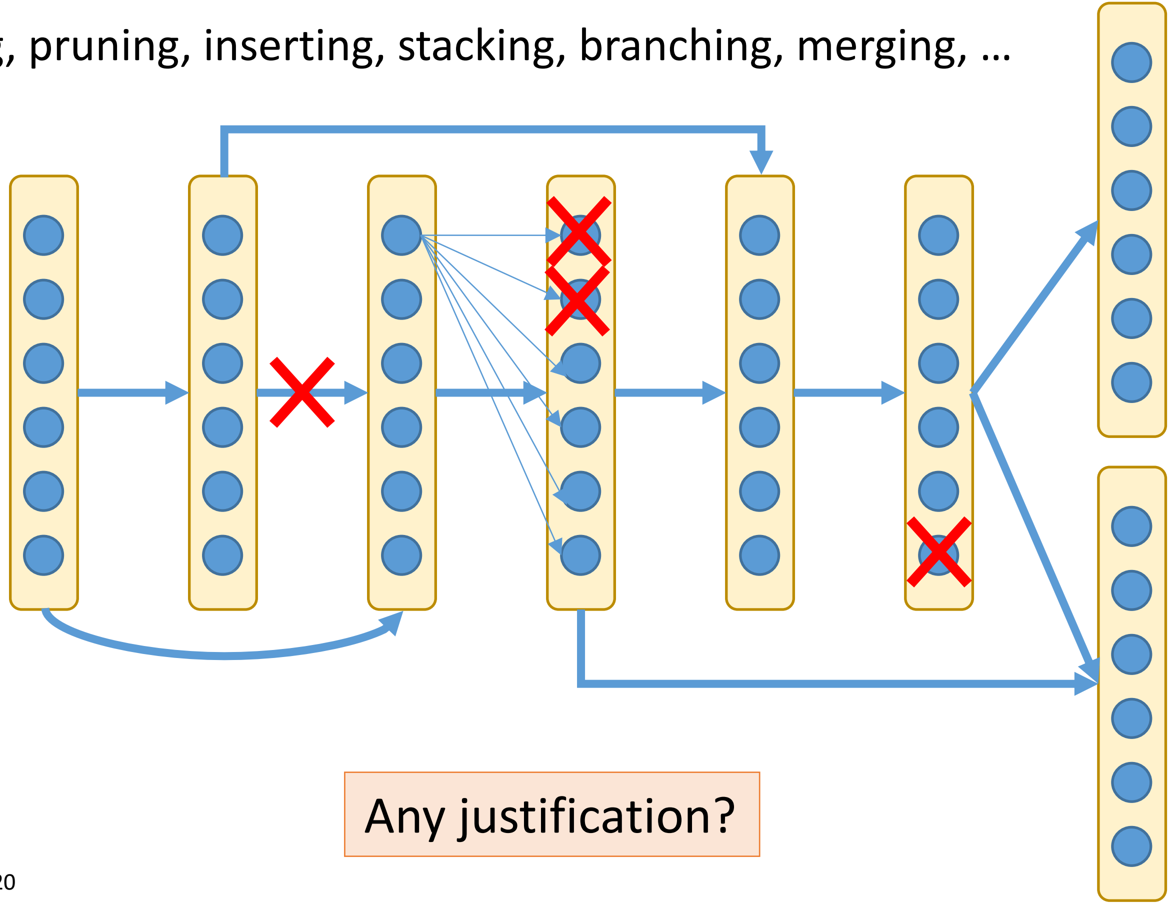






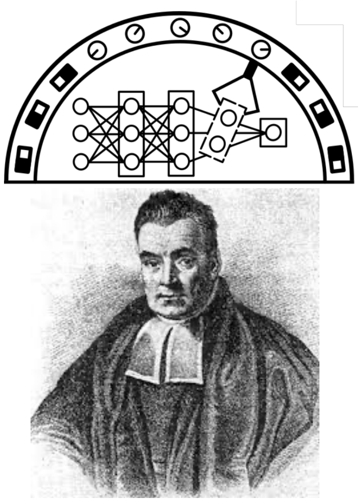
Linking/Unlinking DNN Cells

- Automated neural architecture search may work, but having little justification
- Why linking, pruning, inserting, stacking, branching, merging, …



Any justification?

Part IV



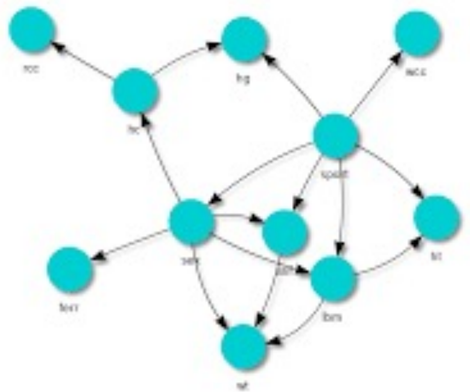
AutoBayes

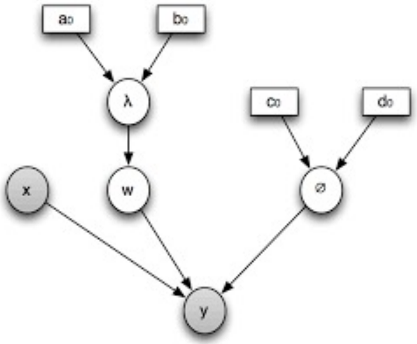
* Different from NASA's *AutoBayes*…

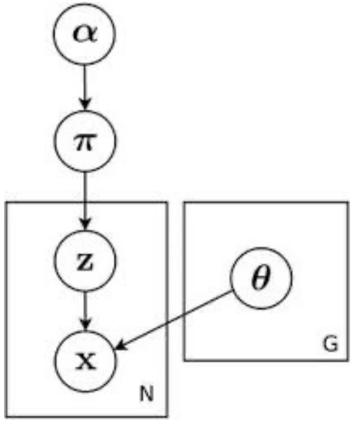
MITSUBISHI ELECTRIC
Changes for the Better

Bayesian Graphical Model

- **Directed graph** indicates marginal dependency





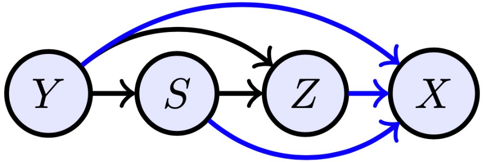


- Joint probability factorization:

$$p(y, s, z, x) = p(y)p(s|y)p(z|s, y)p(x|z, s, y)$$

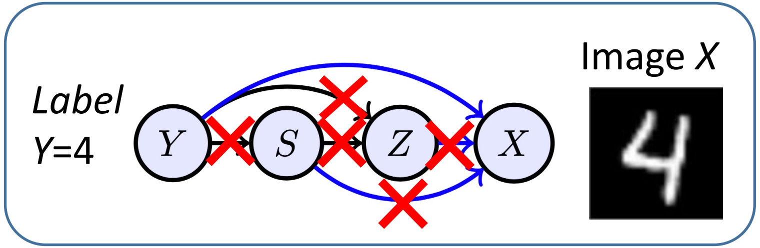
  - ***X***: Measurement (Image, EEG, EMG, …)
  - ***Y***: Label to classify (digit, mental state, …)
  - ***Z***: Latent variables (reduced-dimension feature, …)
  - ***S***: Nuisance variations (user, session, environment, …)
  - Nobody knows true data model…

(4! Possible factorization chains)



Data generative model

Inference Strategy Justified by Bayesian Graph Model



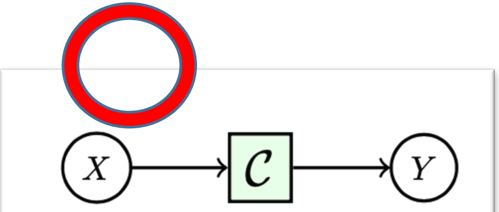
- If true data model follows Markov: ***X* – *Y***

$$p(y)p(s|\cancel{y})p(z|\cancel{s},\cancel{y})p(x|\cancel{z},\cancel{s},y)$$

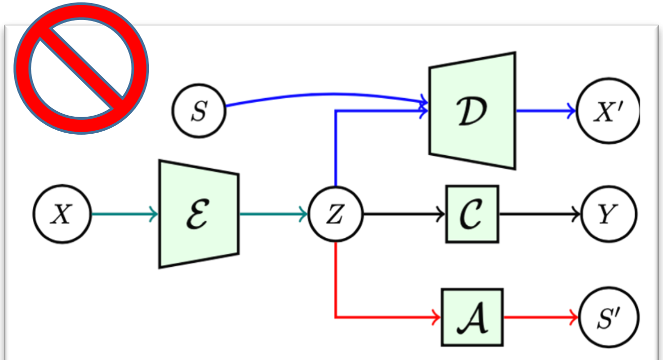
- Likelihood is independent of ***S*** and ***Z***

$$p(y,s,z|x) = p(z|\cancel{x})p(s|\cancel{z},\cancel{x})p(y|\cancel{s},\cancel{z},x)$$

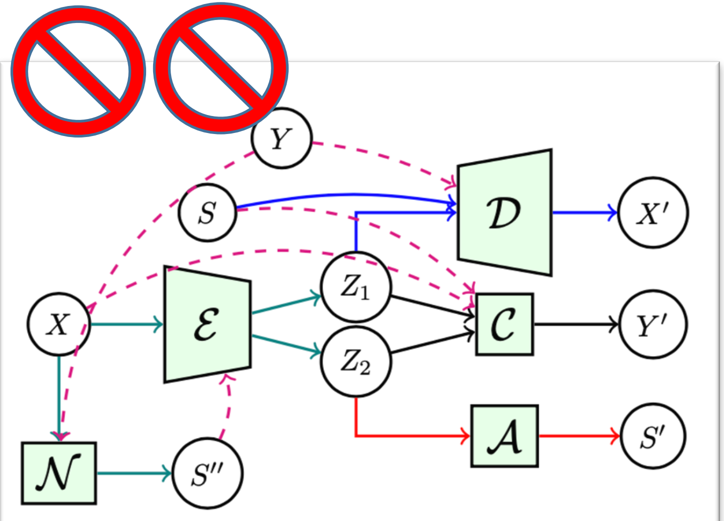
- Simplest classifier model *p*(*y*|*x*) is sufficient
  - *A-CVAE? - No. Irrational to involve more functionality*
  - *I came up with cool complex model. - No, no, no way!*



(a) Standard Classifier Net



(b) Adversarial CVAE-Based Classifier Net

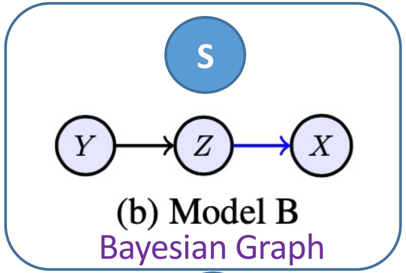


(c) Potentially Extended Classifier Net

Inference Strategy Justified by Bayesian Graph Model (II)

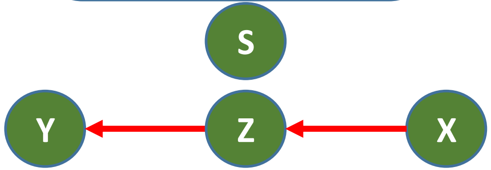
- Consider latent-involved Markov: ***X* – *Z* – *Y***

$$p(y)p(s|\cancel{y})p(z|\cancel{s},y)p(x|z,\cancel{s},\cancel{y})$$



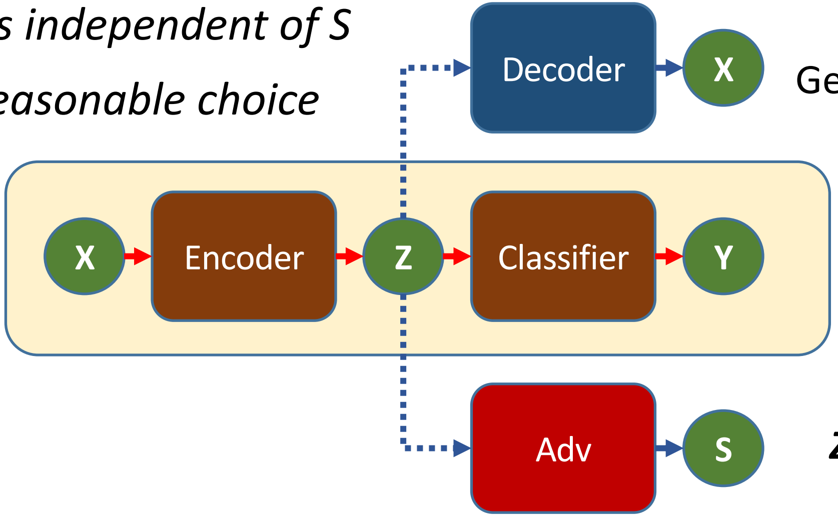
- Then, we have the likelihood

$$p(z|x)p(s|\cancel{z},\cancel{x})p(y|\cancel{s},z,\cancel{x}).$$



Inference Graph

- *I obtained **X-Z-Y** network. – No. Not enough*
- *VAE: I added generative model from Bayesian graph. – Not yet*
- *CVAE? – No, X is independent of S*
- *A-VAE? – Yes, reasonable choice*



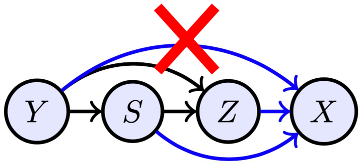
Generative model: VAE

Naïve Inference

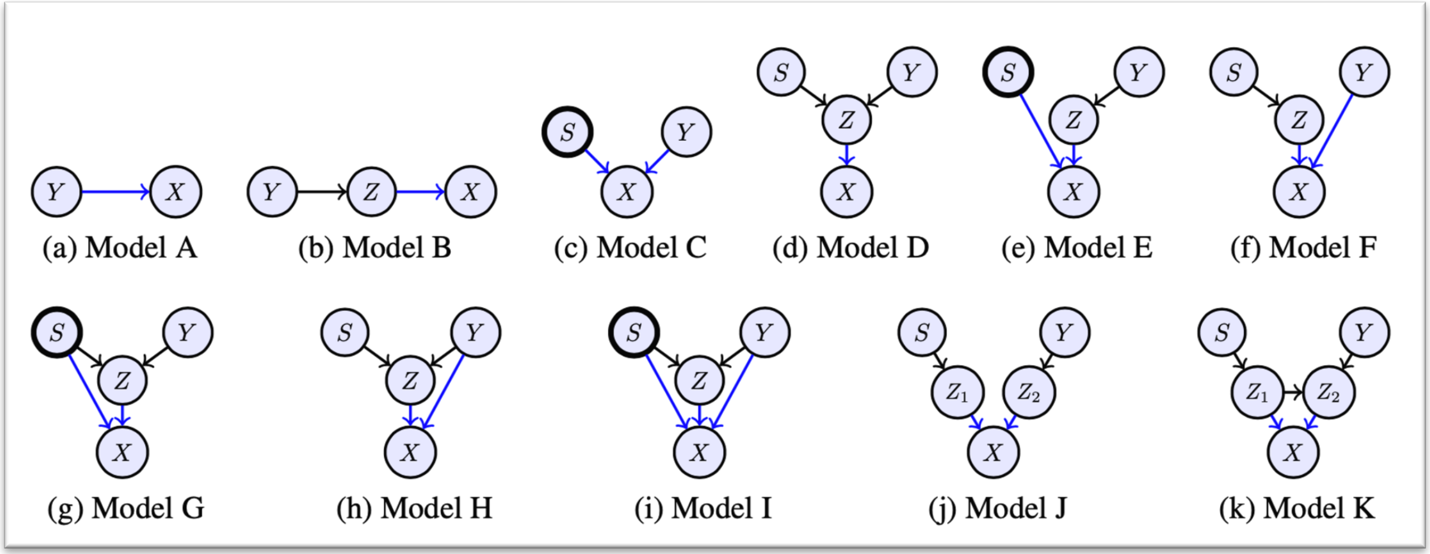
Z is independent of ***S***

Explore Bayesian Graphs to Derive Compact Inference Graphs

- Bayesian graph yields justified inference graphs

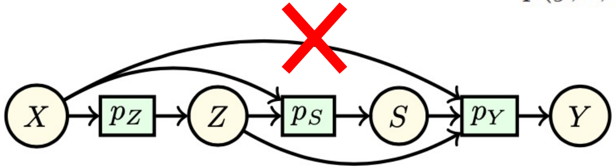


Bayesian graph

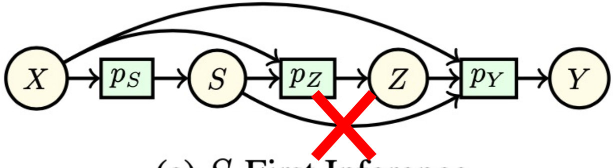


Inference graph

$$p(y,s,z|x) = \begin{cases} p(z|x)p(s|z,x)p(y|s,z,x), & \text{Z-first-inference} \\ p(s|x)p(z|s,x)p(y|z,s,x), & \text{S-first-inference} \end{cases}$$



(b) Z-First Inference



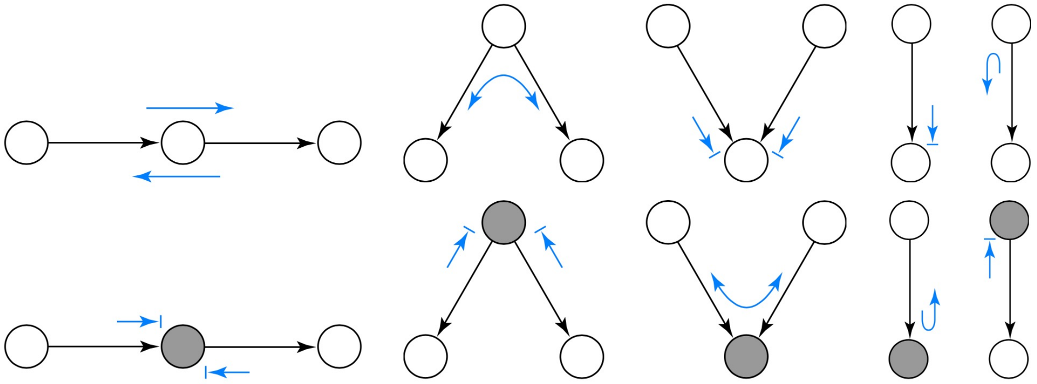
(c) S-First Inference

Redundant links can be identified

Bayes Ball Algorithm

- Conditional independence can be justified systematically with simple **10 rules**

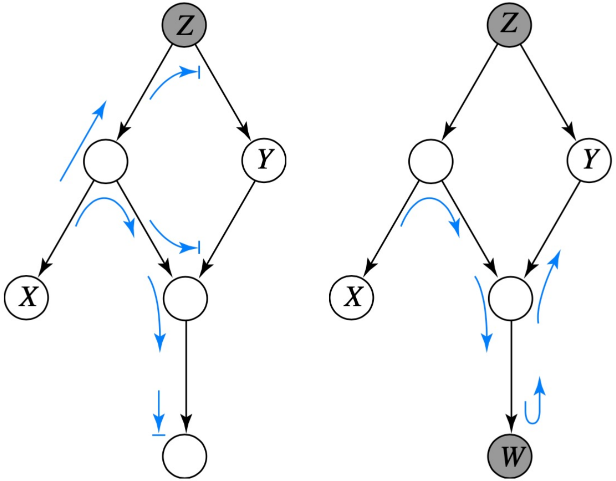
An undirected path is active if a Bayes ball travelling along it never encounters the "stop" symbol: $\longrightarrow\!\!|$



If there are no active paths from X to Y when $\{Z_1, \ldots, Z_k\}$ are shaded, then $X \perp\!\!\!\perp Y \mid \{Z_1, \ldots, Z_k\}$.

Bayes Ball 10 Rules

Example



no active paths

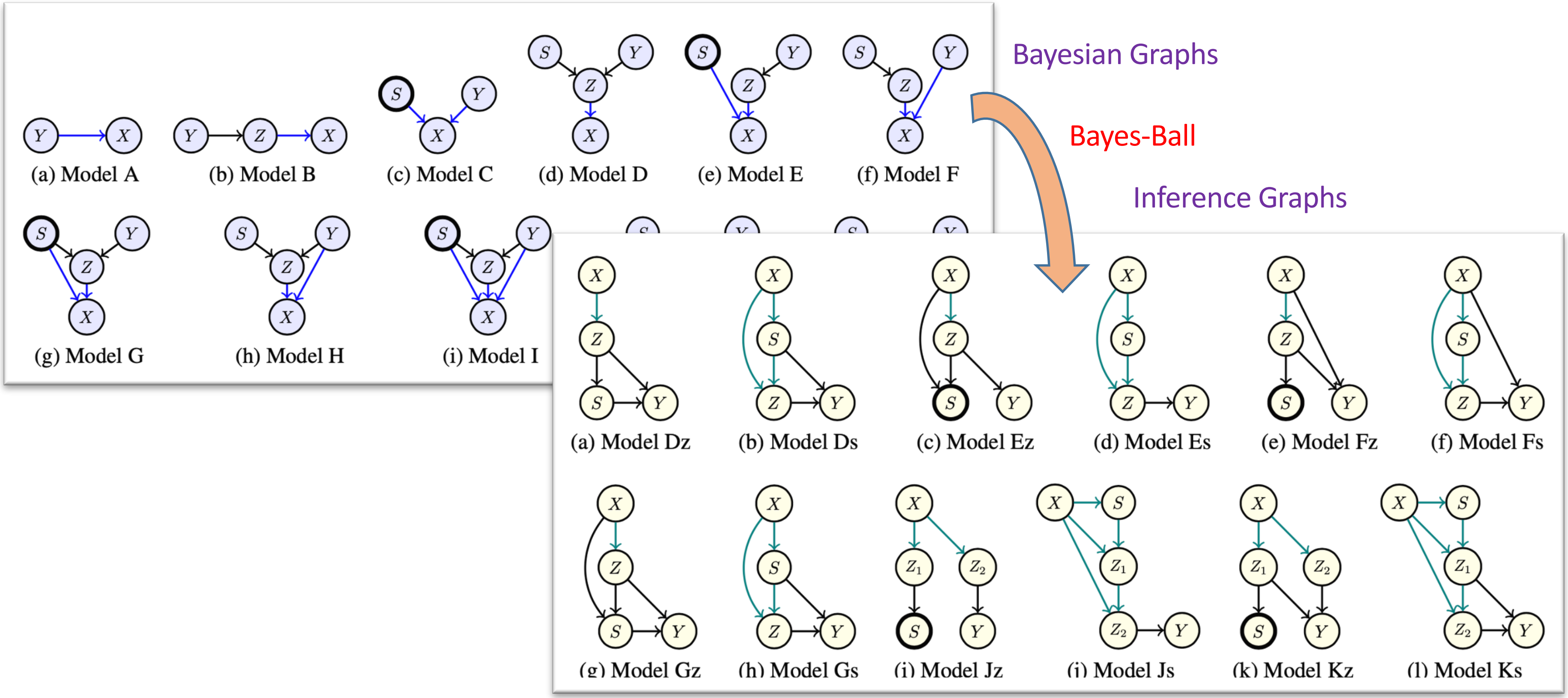
$X \perp\!\!\!\perp Y \mid Z$

one active path

$X \not\perp\!\!\!\perp Y \mid \{W, Z\}$

Bayes-Ball for Compact Inference Graph

- Bayes-Ball finds redundant links for inference graphs

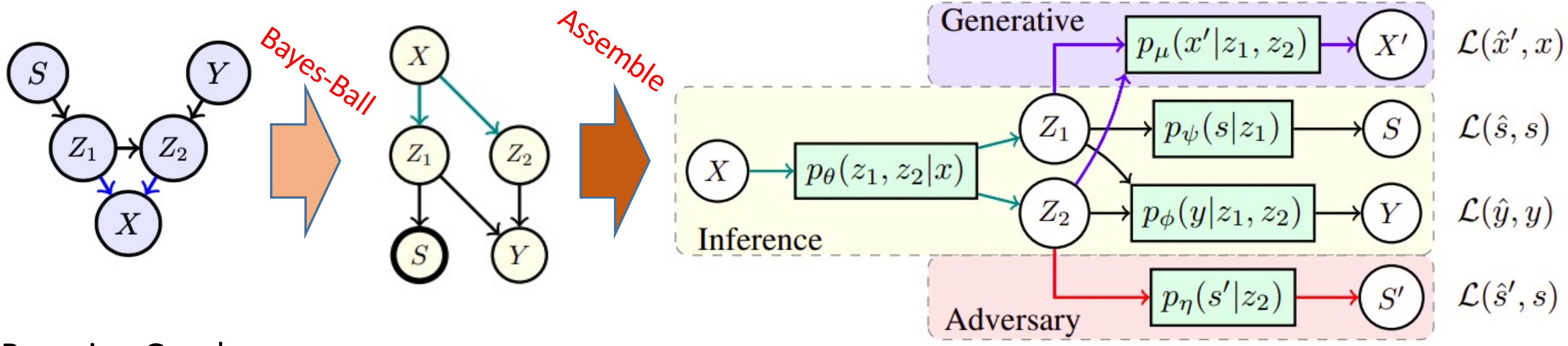


Model Optimization

- Adversarial alternating updates

$$(\theta, \psi, \eta, \mu) = \underset{\theta,\psi,\eta,\mu}{\arg\min}\, \mathbb{E}\big[\mathcal{L}(\hat{y}, y) + \lambda_s \mathcal{L}(\hat{s}, s) + \lambda_x \mathcal{L}(\hat{x}', x) + \lambda_z \mathbb{KL}(z_1, z_2) - \lambda_a \mathcal{L}(\hat{s}', s)\big],$$

$$(z_1, z_2) = p_\theta(x), \quad \hat{y} = p_\psi(z_1, z_2), \quad \hat{s} = p_\phi(z_1), \quad \hat{x}' = p_\mu(z_1), \quad \hat{s}' = p_\eta(z_1, z_2),$$



Bayesian Graph
Model K

Z-Inference Graph

DA-VAE [Han et al. SPL'20]

Proposed AutoBayes Algorithm

- How to connect **Encoder, Decoder, Classifier, Estimator, Adversary** cells?

Algorithm 1 Pseudocode for AutoBayes Framework

Require: Nodes set $\mathcal{V} = [Y, X, S_1, S_2, \ldots, S_n, Z_1, Z_2, \ldots, Z_m]$, where Y denotes task labels, X is a measurement data, $\mathcal{S} = [S_1, S_2, \ldots, S_n]$ are (potentially multiple) semi-supervised nuisance variations, and $\mathcal{Z} = [Z_1, Z_2, \ldots, Z_m]$ are (potentially multiple) latent vectors

Ensure: Semi-supervised training/validation datasets

```
for all permutations of node factorization from Y to X do
    Let B_0 be the corresponding Bayesian graph for the permuted full-chain factorization p(y)···p(z_1|···)···p(x|···)
    for all combinations of link pruning on the full-chain Bayesian graph B_0 do
        Let B be the corresponding pruned Bayesian graph
        Apply the Bayes-Ball algorithm on B to build a conditional independency list I
        for all permutations of node factorization from X to Y do
            Let F_0 be the corresponding factor graph, representing a full-chain conditional probability p(·|x)···p(z_1|···)···p(y|···,x)
            Prune all redundant links in F_0 based on conditional independency I
            Let F be the pruned factor graph
            Merge the pruned Bayesian graph B into the pruned factor graph F
            Attach an adversary network A to latent nodes Z for Z_k ⊥ S ∈ I
            Assign an encoder network E for p(Z|···) in the merged factor graph
            Assign a decoder network D for p(x|···) in the merged factor graph
            Assign a nuisance indicator network N for p(S|···) in the merged factor graph
            Assign a classifier network C for p(y|···) in the merged factor graph
            Adversary train the whole DNN structure with variational reparameterization to minimize a loss function in (11)
        end for                                         ▷ At most (|V| − 2)! combinations
    end for                                             ▷ At most 2^{|V|(|V|−1)/2} combinations
end for                                                 ▷ At most (|V| − 2)! combinations
return the best model having highest task accuracy in validation sets
```

Automatic exploration of Bayesian graphs

Bayes Ball to check independence

Inference model construction

Link Encoder, Decoder, Classifier, Estimator, and Adversary Nets

Return best architectures

MNIST (QMNIST)

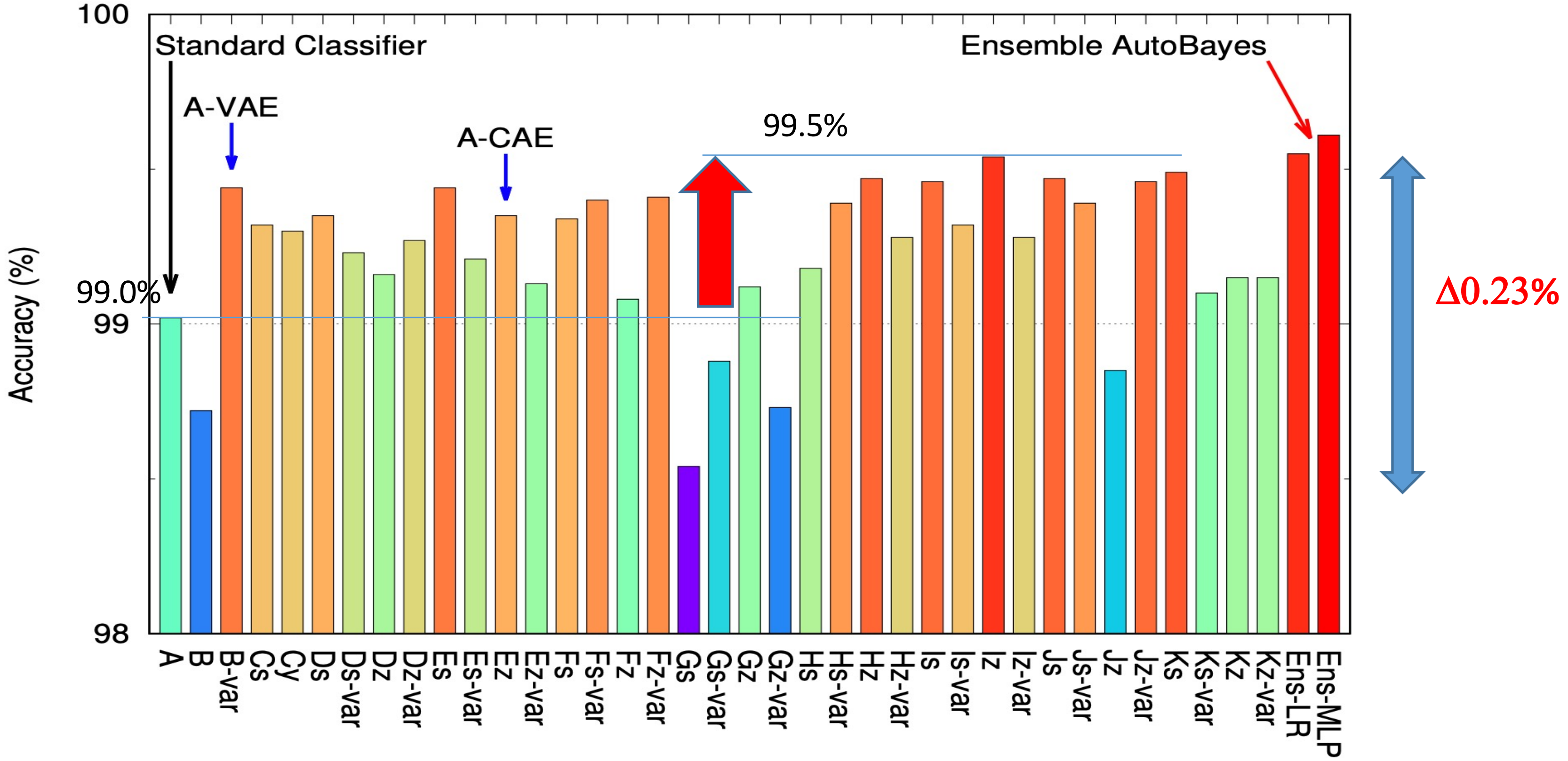
- 28x28 gray-scale images
- 10-class hand-written digits
- 60,000 training data
- 10,000 test data

- Who wrote?
- QMNIST
  https://github.com/facebookresearch/qmnist
  – Identical datasets of MNIST
  – Extended labels (writer ID etc.)
  – Training data were written by **539** NIST employees
  – Testing data were written by **400** high-schoolers

  – **Writer ID** is a nuisance: Hand-written digits may depend on the writer

QMNIST Results

- Up to **0.5% gain** by nuisance-robust inference



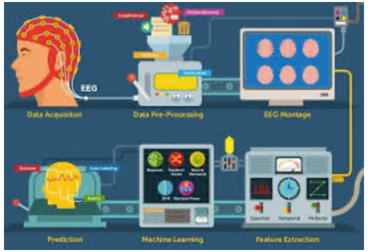
Public Physiological Datasets

- Stress: temperature, **heart rate**, electrodermal activity, arterial oxygen level, etc. for 4-state stress level measurement
- RSVP: **EEG** for rapid serial visual presentation (RSVP) drowsiness test with 4 tasks
- MI: PhysioNet EEG Motor Imagery (MI) dataset with 4-class tasks
- ErrP: An error-related potential (ErrP) of EEG dataset in spelling task
- Faces: An implanted electrocorticography (**ECoG**) array dataset for visual stimulus.
- Ninapro: An electromyogram (**EMG**) dataset for fingers motion detection for prosthetic hands.

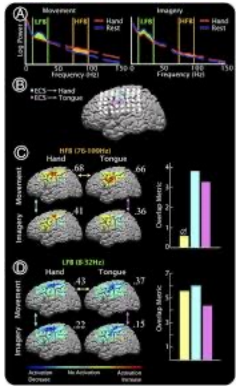
Heart rates

Electrodermal

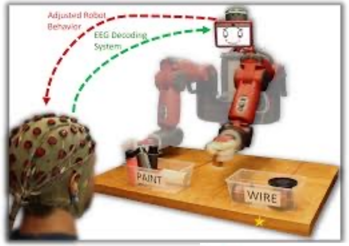
Oxygen



RSVP EEG



MI EEG



ErrP EEG

ECoG

EMG

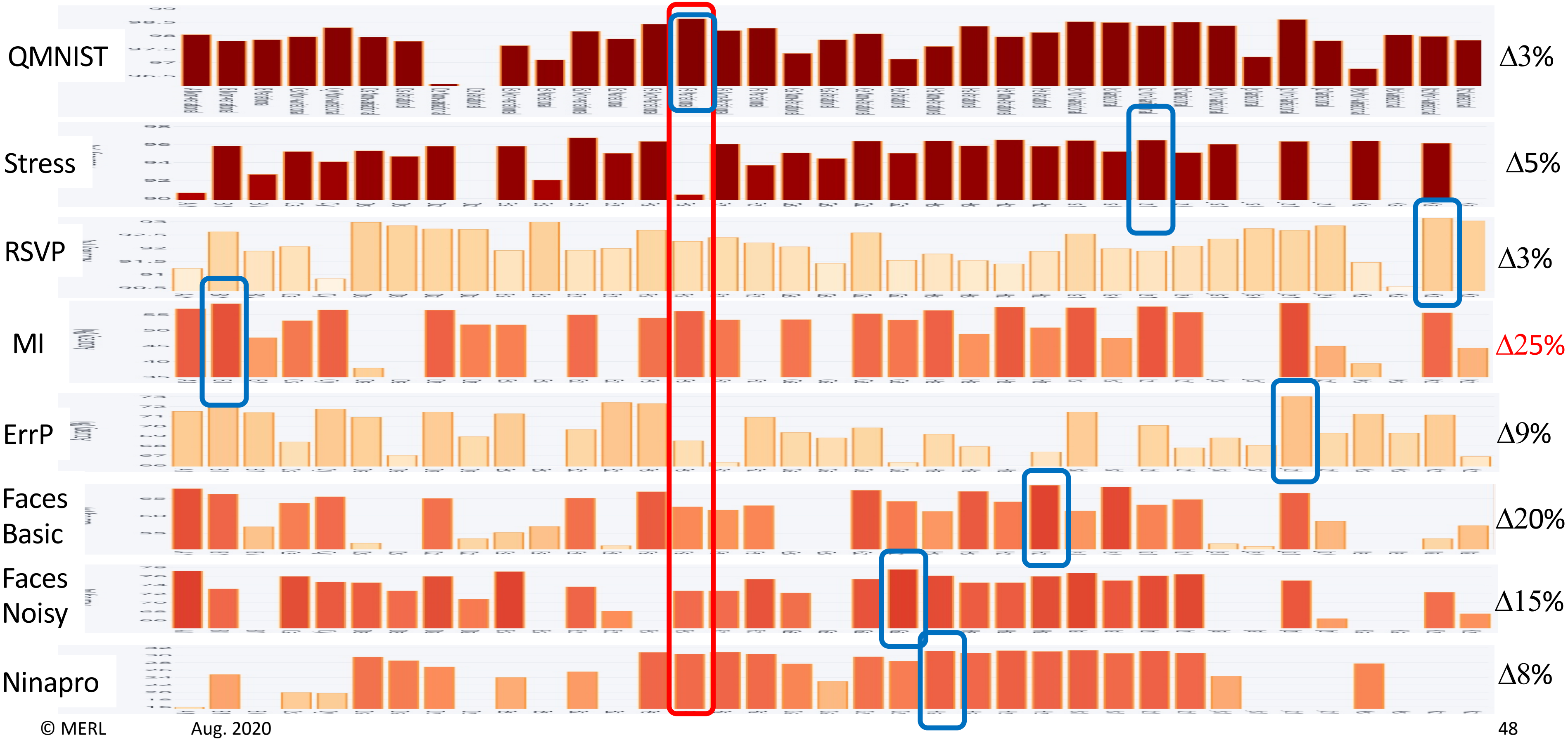
Variety of Datasets

- Publicly available datasets
  - QMNIST: https://github.com/facebookresearch/qmnist
  - Stress: https://physionet.org/content/noneeg/1.0.0/
  - RSVP: http://hdl.handle.net/2047/D20294523
  - MI: https://physionet.org/physiobank/database/eegmmidb/
  - ErrP: https://www.kaggle.com/c/inria-bci-challenge
  - Faces: https://exhibits.stanford.edu/data/catalog/zk881ps0522
  - Ninapro: https://zenodo.org/record/1000116#.XuIppS2z3OR

Datasets	Modality	Dimension	Nuisance ($\|S\|$)	Labels ($\|Y\|$)	Samples
QMNIST	Image	28×28	539	10	60,000
Stress	Temperature etc.	7	20	4	24,000
RSVP	EEG	16×128	10	4	41,400
MI	EEG	64×480	106	4	9,540
ErrP	EEG	56×250	27	2	9,180
Faces Basic	ECoG	31×400	14	2	4,100
Faces Noisy	ECoG	39×400	7	2	2,100
Ninapro	EMG	16	10	12	890,446

AutoBayes Benefit: Explore Different Models for Different Problems

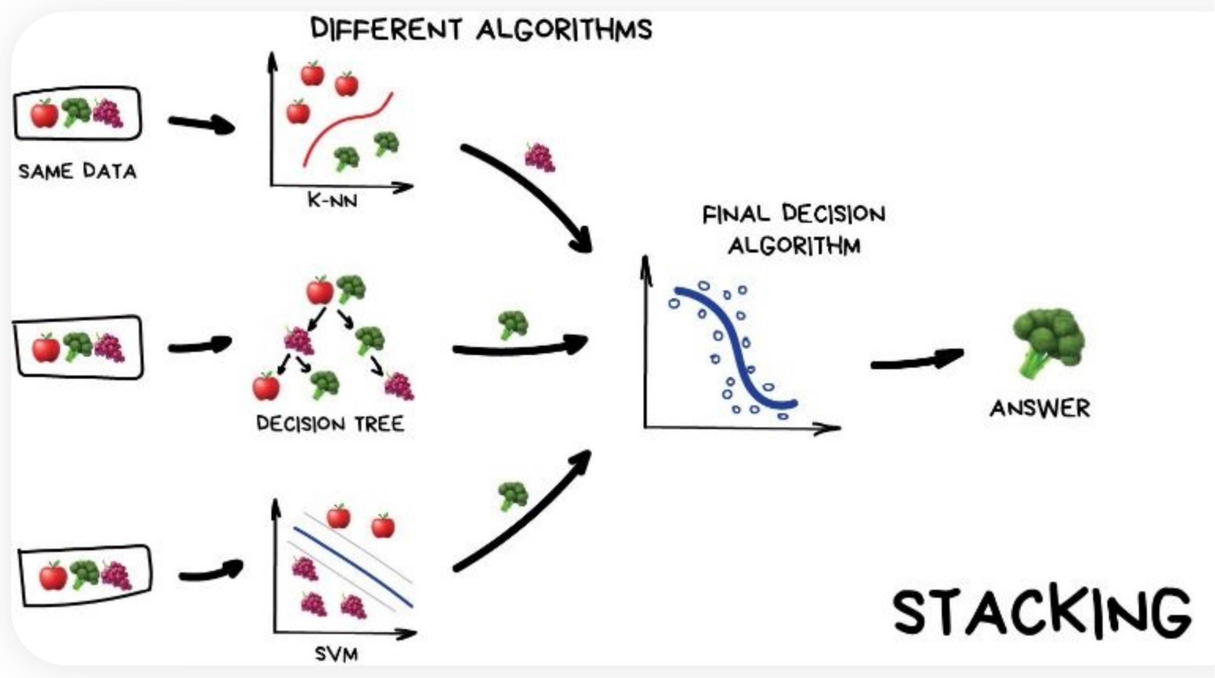
- *No Free Lunch Theorems*: There is no one model that performs best for every dataset



Part V

Ensemble Learning

- Every single model may be weak
- **Combining multiple weak models** may beat one strong model

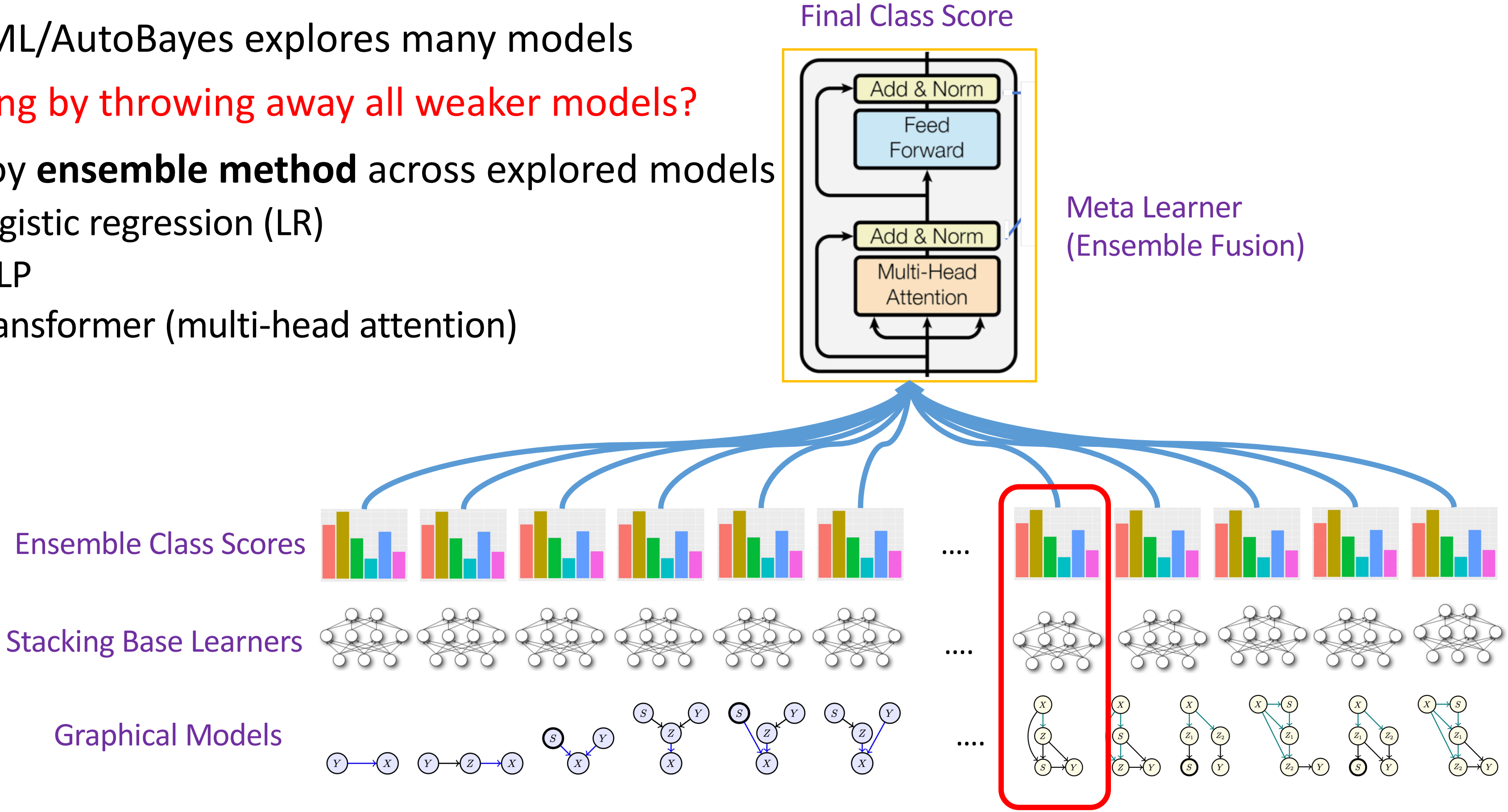


Tiny weak fishes

Gigantic strong fish

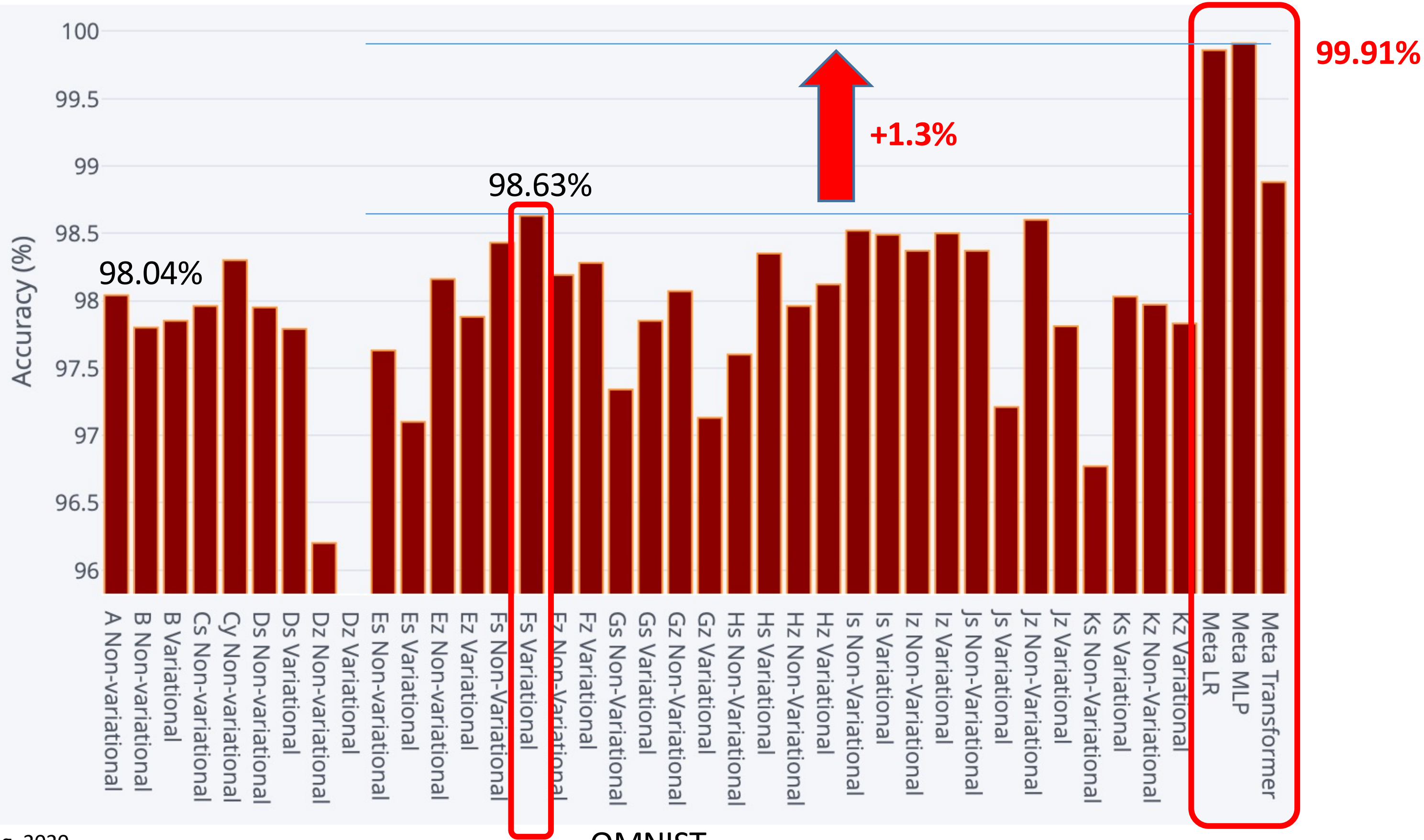
Ensemble Aggregation

- AutoML/AutoBayes explores many models
- Wasting by throwing away all weaker models?
- Employ **ensemble method** across explored models
  - Logistic regression (LR)
  - MLP
  - Transformer (multi-head attention)
  - …



Ensemble Leaning Gain in AutoBayes (QMNIST)

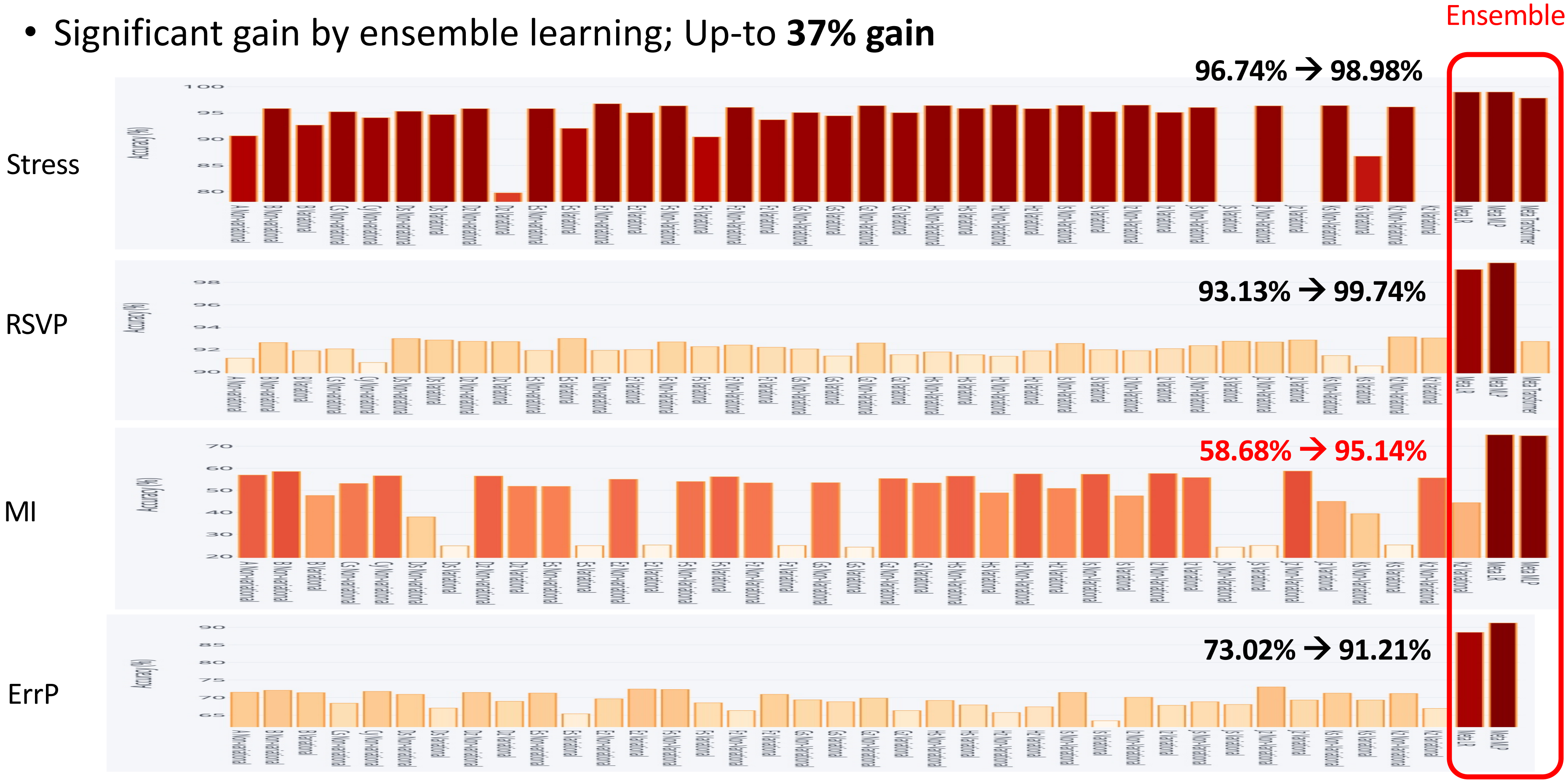
- Significant gain by ensemble learning; **1.3% gain**, state-of-the-art accuracy



QMNIST

Ensemble Leaning Gain in AutoBayes (heartrate, EEG)

- Significant gain by ensemble learning; Up-to **37% gain**

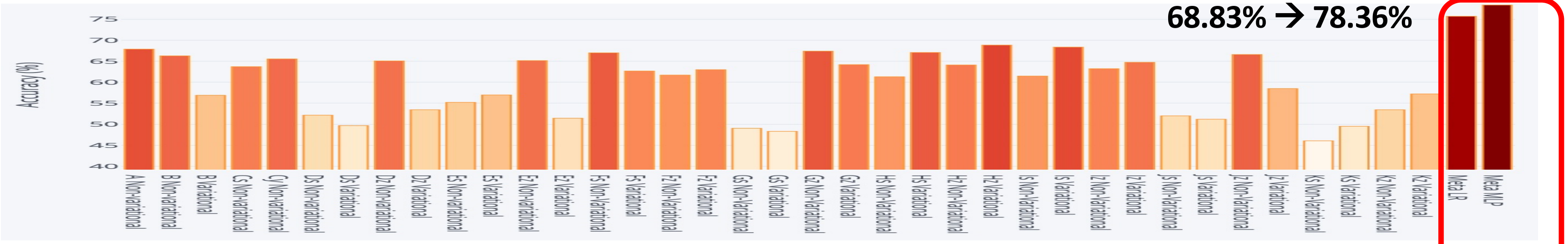


Ensemble Leaning Gain in AutoBayes (ECoG, EMG)

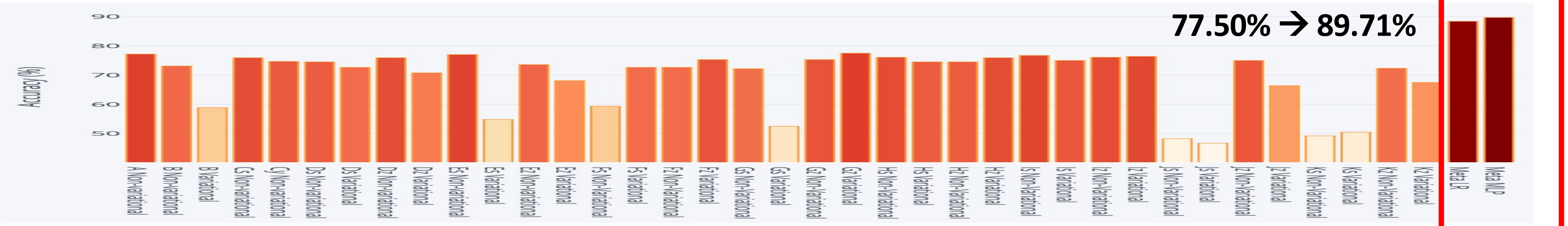
- Significant gain by ensemble learning; Up-to **12% gain**

Ensemble

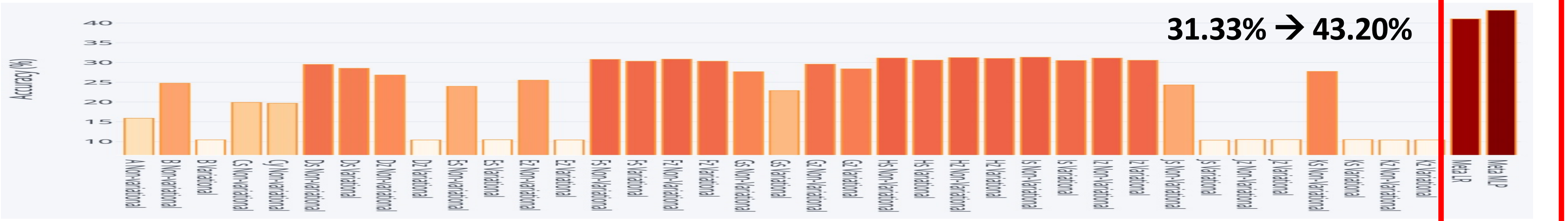
Faces Basic



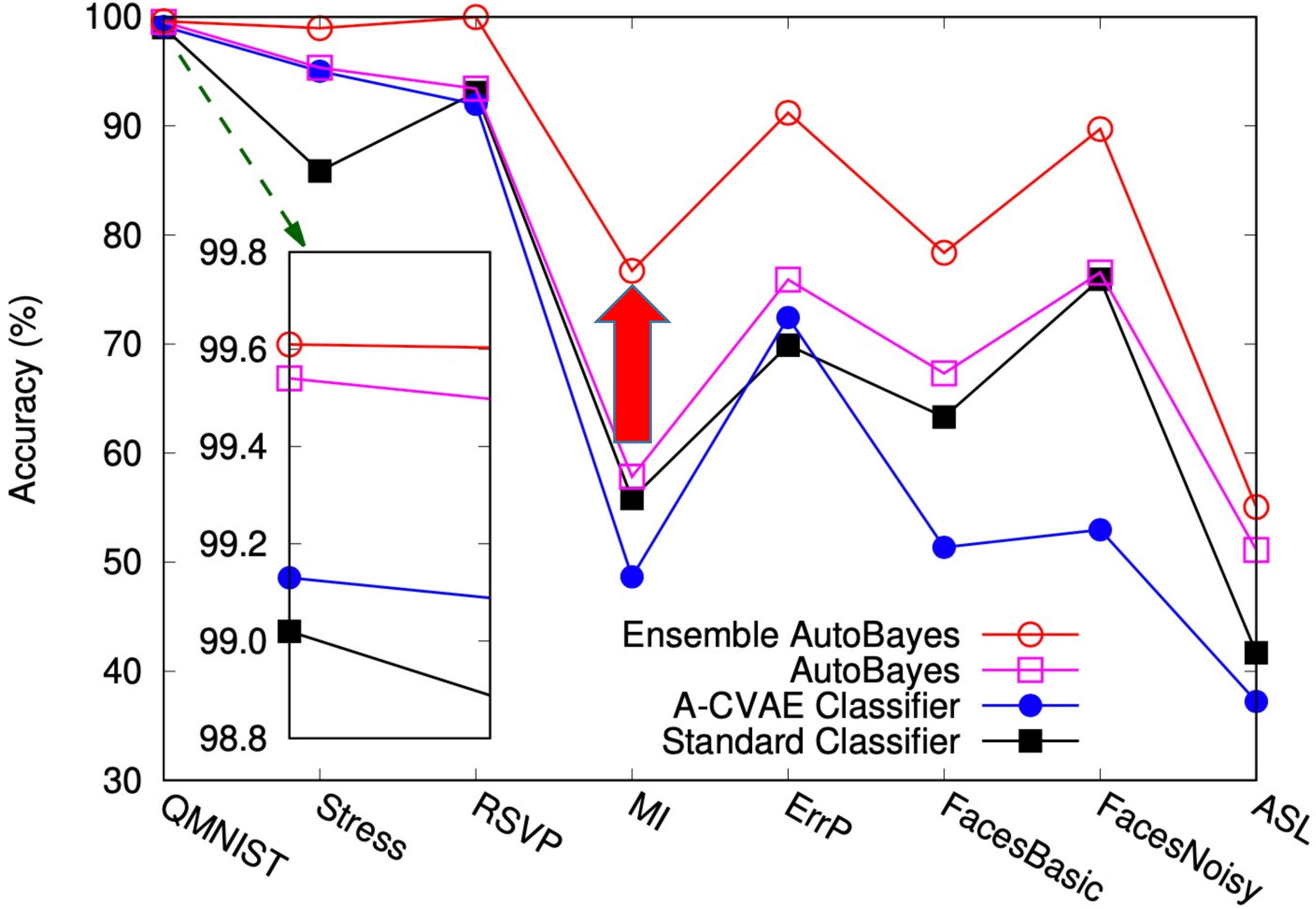
Faces Noisy



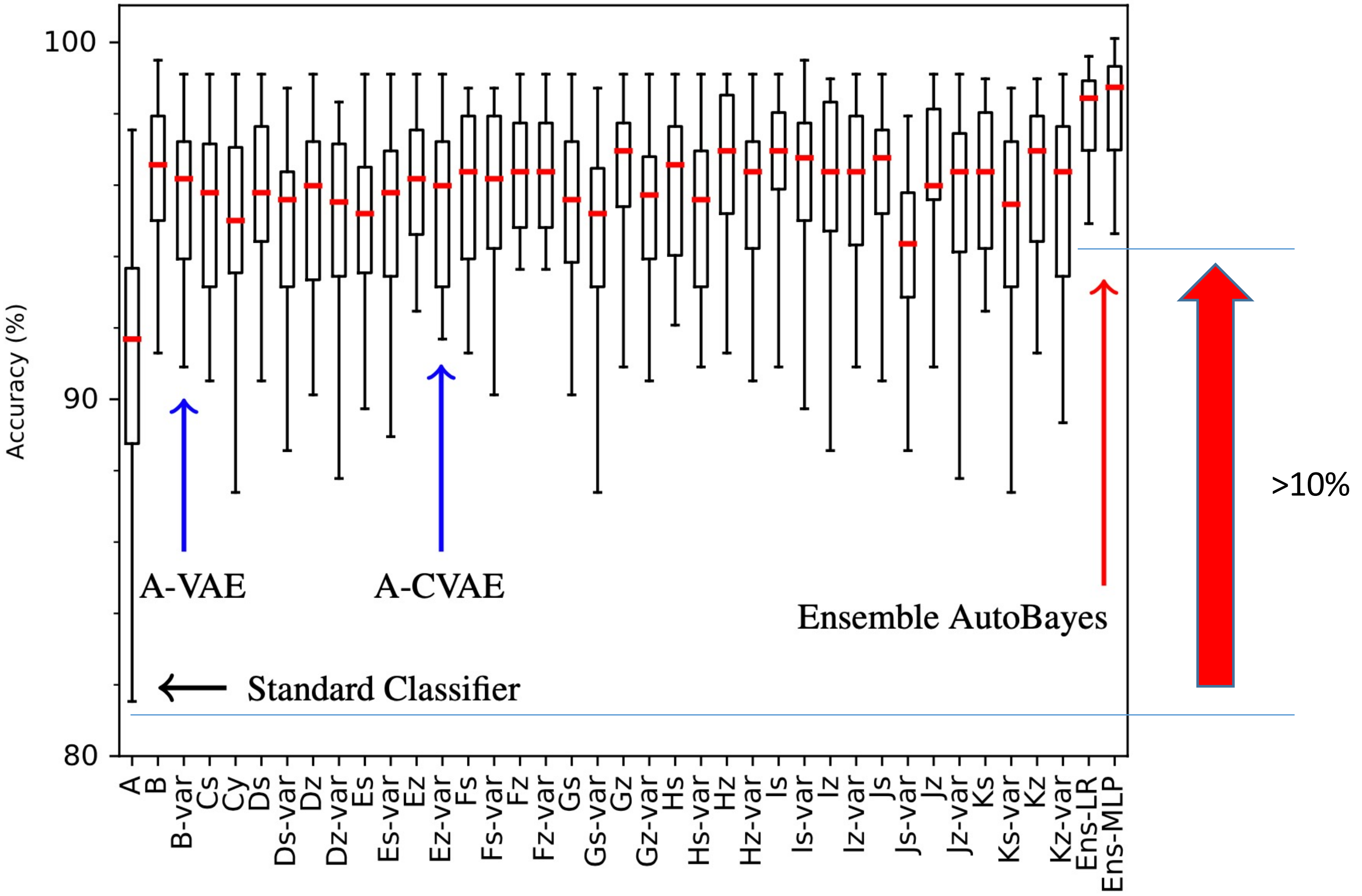
Ninapro



AutoBayes Ensemble Gain



Subject Variation Robustness (Stress Dataset)

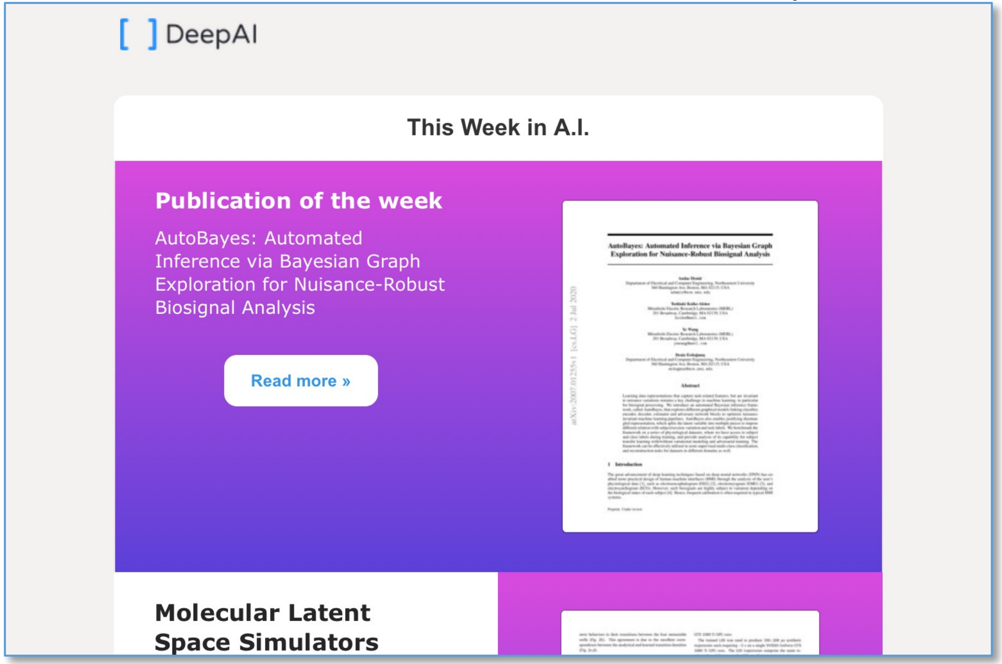


Summary

- We introduced a new concept called **AutoBayes** for macro DNN architecture exploration
  - Different **Bayesian graphs** are explored systematically
  - **Bayes Ball** algorithm justifies pruning independent edges
  - Encoder, decoder, estimator, classifier, and adversary network blocks are rationally linked
- We also discussed **transfer learning, adversarial learning, ensemble learning**
  - Multiple architectures explored in AutoML are not wasted for final classification (as **base learners**)
  - Different **meta learners** (LR, MLP, Transformer) are evaluated to aggregate multiple models
- Demonstrated the benefit for various public physiological datasets
  - Various different modalities (**image, heartrate, EEG, ECoG, EMG**) and dimensionalities are considered
- Questions?
  - Contact us: koike@merl.com, yewang@merl.com
  - More details in arXiv: https://arxiv.org/abs/2007.01255

DeepAI

- https://deepai.org/publication/autobayes-automated-inference-via-bayesian-graph-exploration-for-nuisance-robust-biosignal-analysis
- When our arXiv was uploaded on July 2, it became *top trending paper*
  - Obtained **71 “likes” in 4 days**
  - It was highlighted in **“This Week in A.I.” Newsletter** on July 11

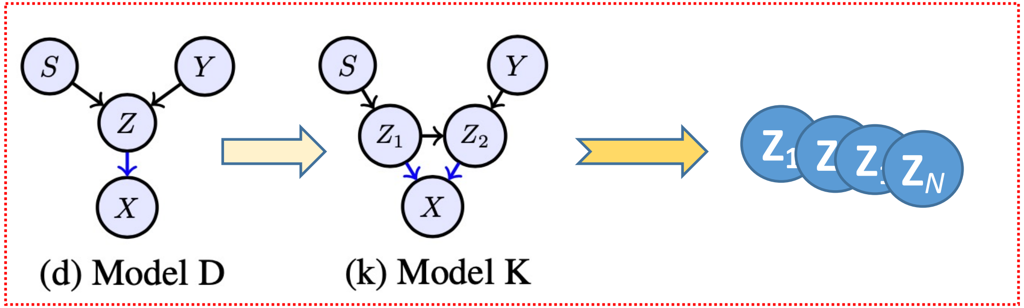


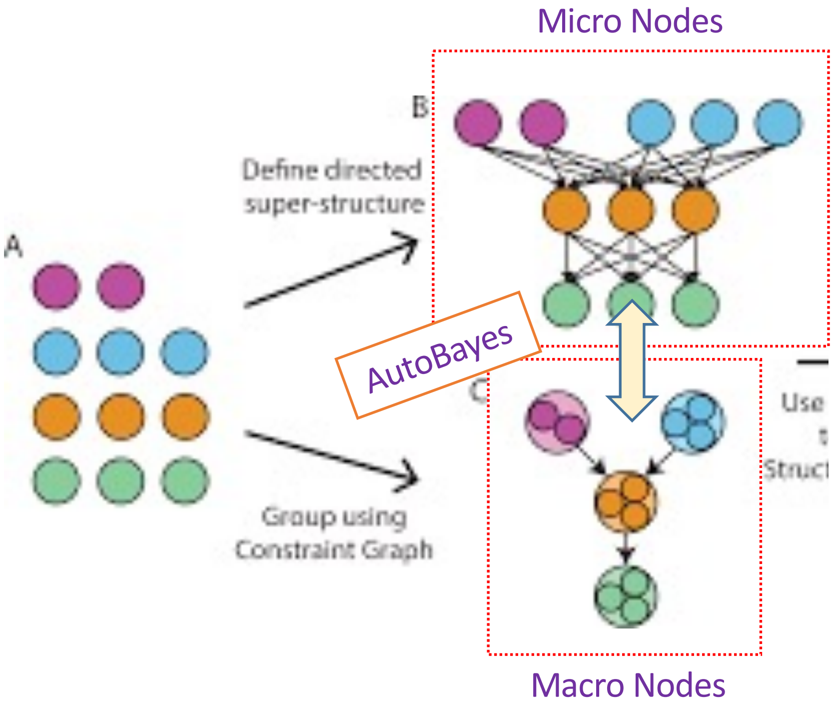
Contributions

- AutoBayes explores potential graphical models inherent to the data, rather than exploring hyperparameters of DNN blocks.
- AutoBayes offers a solid reason of how to connect multiple DNN blocks to impose conditioning and adversary censoring for the task classifier, feature encoder, decoder, nuisance indicator and adversary networks, based on an explored Bayesian graph.
- It provides a systematic automation framework to explore different inference models through the use of the Bayes-Ball algorithm and ordered factorization.
- The framework is also extensible to multiple latent representations and multiple nuisance factors.
- Besides fully-supervised training, AutoBayes can automatically build some relevant graphical models suited for semi-supervised learning.
- Ensemble learning is introduced to improve performance while AutoBayes model exploration

AutoBayes as AutoML: Macro to Micro Exploration

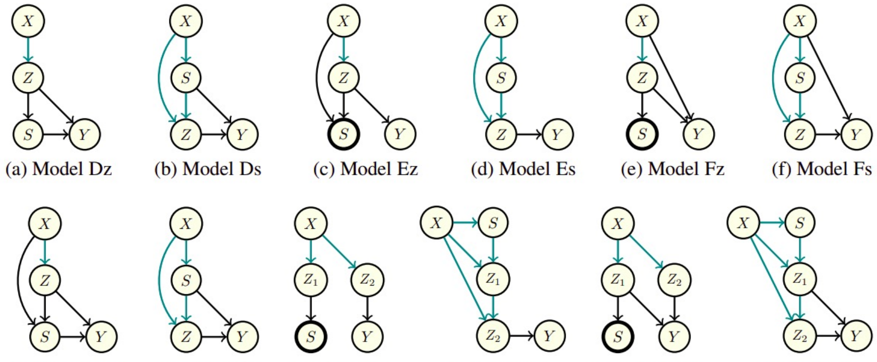
- In principle, AutoBayes can be applied to arbitrary number of nodes
- Splitting X, Y, Z, S macro-nodes into scalar-valued micro-nodes, AutoBayes operates like AutoML architecture search but with more theoretical justification





Methodology

$$
p(y,s,z,x) = \begin{cases}
p(y)p(s|\cancel{y})p(z|\cancel{s},\cancel{y})p(x|\cancel{z},\cancel{s},y), & \text{Model-A}\\
p(y)p(s|\cancel{y})p(z|\cancel{s},y)p(x|z,\cancel{s},\cancel{y}), & \text{Model-B}\\
p(y)p(s|\cancel{y})p(z|\cancel{s},\cancel{y})p(x|\cancel{z},s,y), & \text{Model-C}\\
p(y)p(s|\cancel{y})p(z|s,y)p(x|z,\cancel{s},\cancel{y}), & \text{Model-D}\\
p(y)p(s|\cancel{y})p(z|\cancel{s},y)p(x|z,s,\cancel{y}), & \text{Model-E}\\
p(y)p(s|\cancel{y})p(z|s,\cancel{y})p(x|z,\cancel{s},y), & \text{Model-F}\\
p(y)p(s|\cancel{y})p(z|s,y)p(x|z,s,\cancel{y}), & \text{Model-G}\\
p(y)p(s|\cancel{y})p(z|s,y)p(x|z,\cancel{s},y), & \text{Model-H}\\
p(y)p(s|\cancel{y})p(z|s,y)p(x|z,s,y), & \text{Model-I}\\
p(y)p(s|\cancel{y})p(z_1|s,\cancel{y})p(z_2|\cancel{z_1},\cancel{s},y)p(x|z_2,z_1,\cancel{s},\cancel{y}), & \text{Model-J}\\
p(y)p(s|\cancel{y})p(z_1|s,\cancel{y})p(z_2|z_1,\cancel{s},y)p(x|z_2,z_1,\cancel{s},\cancel{y}), & \text{Model-K}
\end{cases}
$$



(a) Model Dz (b) Model Ds (c) Model Ez (d) Model Es (e) Model Fz (f) Model Fs

(g) Model Gz (h) Model Gs (i) Model Jz (j) Model Js (k) Model Kz (l) Model Ks

- Slash-cancelled factors from the full-chain case explicitly indicate independence.
- Conditional independence enables pruning links in the inference factor graphs.

$$
p(y,z_1,z_2,s|x) = \begin{cases}
p(z_1,z_2|x)p(y,s|z_1,z_2,\cancel{x}), & \text{z/ys}\\
p(z_1,z_2|x)p(s|z_1,\cancel{z_2},\cancel{x})p(y|\cancel{s},\cancel{z_1},z_2,\cancel{x}), & \text{Z-Inference}\\
p(z_1|x)p(z_2|z_1,x)p(s|z_1,\cancel{z_2},\cancel{x})p(y|\cancel{s},\cancel{z_1},z_2,\cancel{x}), & \text{z2/z1/s/y}\\
p(z_2|x)p(z_1|z_2,x)p(s|z_1,\cancel{z_2},\cancel{x})p(y|\cancel{s},\cancel{z_1},z_2,\cancel{x}), & \text{z1/z2/s/y}\\
p(z_1|x)p(s|z_1,x)p(z_2|s,z_1,x)p(y|\cancel{s},\cancel{z_1},z_2,\cancel{x}), & \text{z2/s/z1/y}\\
p(s|x)p(z_1|s,x)p(z_2|s,z_1,x)p(y|\cancel{s},\cancel{z_1},z_2,\cancel{x}), & \text{s/z2/z1/y}\\
p(z_1|x)p(s|z_1,\cancel{x})p(z_2|\cancel{s},z_1,x)p(y|\cancel{s},z_2,\cancel{z_1},\cancel{x}), & \text{z1/s/z2/y}\\
p(s|x)p(z_1|s,x)p(z_2|\cancel{s},z_1,x)p(y|\cancel{s},z_2,\cancel{z_1},\cancel{x}), & \text{s/z1/z2/y}\\
p(s|x)p(z_1,z_2|s,x)p(y|\cancel{s},z_2,\cancel{z_1},\cancel{x}), & \text{S-Inference}\\
\ldots
\end{cases} \tag{20}
$$

Z-first and S-first inference graph models relevant for generative models D–G, J, and K

MITSUBISHI ELECTRIC

Changes for the Better