AUTOBAYES: AUTOMATED BAYESIAN GRAPH EXPLORATION FOR NUISANCE-ROBUST INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

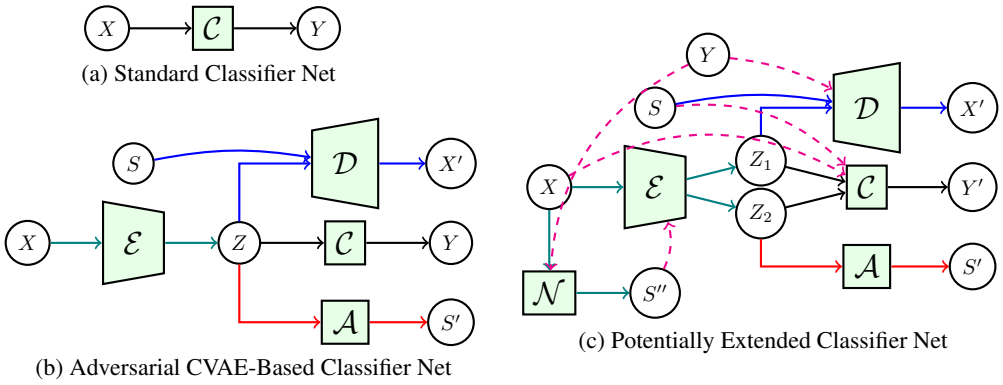
Learning data representations that capture task-related features, but are invariant to nuisance variations remains a key challenge in machine learning. We introduce an automated Bayesian inference framework, called AutoBayes, that explores different graphical models linking classifier, encoder, decoder, estimator and adversary network blocks to optimize nuisance-invariant machine learning pipelines. AutoBayes also enables learning disentangled representations, where the latent variable is split into multiple pieces to impose different relation with nuisance variation and task labels. We benchmark the framework on several public datasets, where we have access to subject and class labels during training, and provide analysis of its capability for subject-transfer learning with/without variational modeling and adversarial training. We demonstrate a significant performance improvement by ensemble stacking across explored graphical models.

1 INTRODUCTION

The great advancement of deep learning techniques based on deep neural networks (DNN) has enabled more practical design of human-machine interfaces (HMI) through the analysis of the user's physiological data (Faust et al., 2018), such as electroencephalogram (EEG) (Lawhern et al., 2018) and electromyogram (EMG) (Atzori et al., 2016). However, such biosignals are highly subject to variation depending on the biological states of each subject (Christoforou et al., 2010). Hence, frequent calibration is often required in typical HMI systems.

Toward resolving this issue, subject-invariant methods (Özdenizci et al., 2019b), employing adversarial training (Makhzani et al., 2015; Lample et al., 2017; Creswell et al., 2017) with Conditional Variational AutoEncoder (A-CVAE) (Louizos et al., 2015; Sohn et al., 2015) shown in Fig. 1(b), have emerged to reduce user calibration for realizing successful HMI systems. Compared to a standard DNN classifier $\mathcal{C}$ in Fig. 1(a), integrating additional functional blocks for encoder $\mathcal{E}$, nuisance-conditional decoder $\mathcal{D}$, and adversary $\mathcal{A}$ networks offers excellent subject-invariant performance. The DNN structure may be potentially extended with more functional blocks and more latent nodes as shown in Fig. 1(c). However, most works rely on human design and insight to determine the block connectivity and architecture of DNNs. Automation of hyperparameter and architecture exploration in the context of AutoML (Ashok et al., 2017; Brock et al., 2017; Cai et al., 2017; He et al., 2018; Miikkulainen et al., 2019; Real et al., 2017; 2020; Stanley & Miikkulainen, 2002; Zoph et al., 2018) can facilitate DNN design suited for nuisance-invariant inference. Nevertheless, without proper reasoning, most of the search space for link connectivity will be pointless.

In this paper, we propose a systematic automation framework called AutoBayes, which searches for the best inference graph model associated to a Bayesian graph model well-suited to reproduce the training datasets. The proposed method automatically formulates various different Bayesian graphs by factorizing the joint probability distribution in terms of data, class label, subject identification (ID), and inherent latent representations. Given Bayesian graphs, some meaningful inference graphs are generated through the Bayes-Ball algorithm (Shachter, 2013) for pruning redundant links to achieve high-accuracy estimation. In order to promote robustness against nuisance parameters such as subject IDs, the explored Bayesian graphs can provide reasoning to use adversarial training with/without variational modeling and latent disentanglement. We demonstrate that AutoBayes can achieve excellent performance across various public datasets, and in particular with an ensemble stacking of multiple explored graphical models.



(a) Standard Classifier Net

(b) Adversarial CVAE-Based Classifier Net

(c) Potentially Extended Classifier Net

Figure 1: Inference methods to classify Y given data X under latent Z and semi-labeled nuisance S.

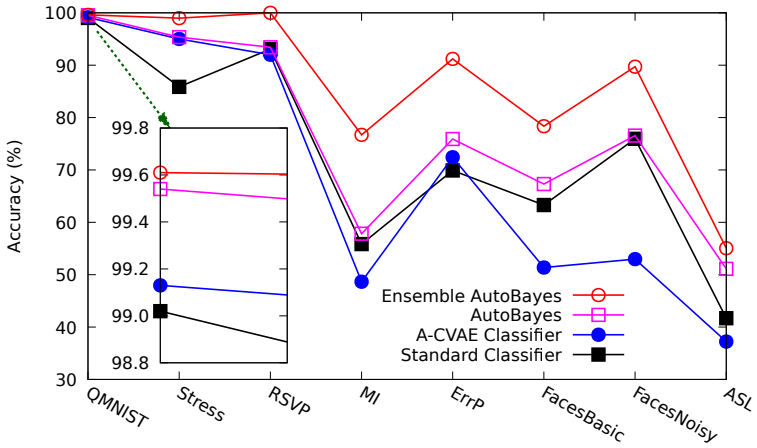


Figure 2: Model accuracy across different datasets. AutoBayes offers significant gain.

2 Key Contributions

At the core of our methodology is the consideration of graphical Bayesian models that capture the probabilistic relationship between random variables representing the data features X, task labels Y, nuisance variation labels S, and (potential) latent representations Z. The ultimate goal is to infer the task label Y from the measured data feature X, which is hindered by the presence of nuisance variations (e.g., inter-subject/session variations) that are (partially) labelled by S. One may use a standard DNN to classify Y given X as shown in Fig. 1(a), without explicitly involving S or Z. Although A-CVAE in Fig. 1(b) may offer nuisance-robust performance through adversarial disentanglement of S from latent Z, there is no guarantee that such a model can perform well across different datasets. It is exemplified in Fig. 2 where A-CVAE outperforms the standard DNN model for some datasets (QMNIST, Stress, ErrP) while it does not for the other cases. It may be because the probabilistic relation underlying data varies over datasets. Our proposed framework can construct justifiable models, achieving higher performance for every dataset, as demonstrated in Fig. 2. It is verified that significant gain can be realized with ensemble methods of different Bayesian graphs which are explored in our AutoBayes. For example, our method with a relatively shallow architecture achieves 99.61% accuracy which is close to state-of-the-art performance in QMNIST dataset.

The main contributions of this paper over the existing works are six-fold as follows:

- AutoBayes explores potential graphical models inherent to the data by combinatorial pruning of dependency assumptions (edges) and then applies Bayes-Ball to examine various inference strategies, rather than blindly exploring hyperparameters of DNN blocks.

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
    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 factor graph corresponding to a full-chain conditional probability
            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 to minimize a loss function in (5)
        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
```

- 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 nuisances factors.
- Besides fully-supervised training, AutoBayes can automatically build some relevant graphical models suited for semi-supervised learning.
- Multiple graphical models explored in AugoBayes can be efficiently exploited to improve performance by ensemble stacking.

We note that this paper relates to some existing literature in AutoML, variational Bayesian inference (Kingma & Welling, 2013; Sohn et al., 2015; Louizos et al., 2015), and adversarial training (Goodfellow et al., 2014; Dumoulin et al., 2016; Donahue et al., 2016; Makhzani et al., 2015; Lample et al., 2017; Creswell et al., 2017), as addressed in Appendix A.1. Nonetheless, AutoBayes is a novel framework that diverges from AutoML, which is mostly employed to architecture tuning at a micro level. Our work focuses on exploring neural architectures at a macro level, which is not an arbitrary diversion, but a necessary interlude. To make further progress in AutoML, our method focuses on the relationships between the connections in a neural network's architecture and the characteristics of the data that is being presented to it (Minsky & Papert, 2017).

3 AUTOBAYES

AutoBayes Algorithm: The overall procedure of the AutoBayes algorithm is described in the pseudocode of Algorithm 1. The AutoBayes automatically constructs non-redundant inference factor graphs given a hypothetical Bayesian graph assumption, through the use of the Bayes-Ball algorithm. Depending on the derived conditional independency and pruned factor graphs, DNN blocks for encoder $\mathcal{E}$, decoder $\mathcal{D}$, classifier $\mathcal{C}$, nuisance estimator $\mathcal{N}$ and adversary $\mathcal{A}$ are reasonably connected. The whole DNN blocks are trained with adversary learning in a variational Bayesian inference.

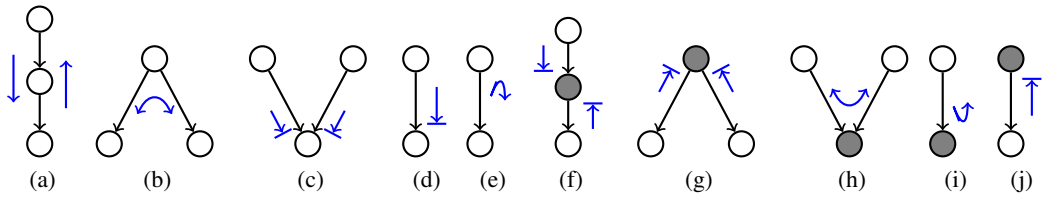
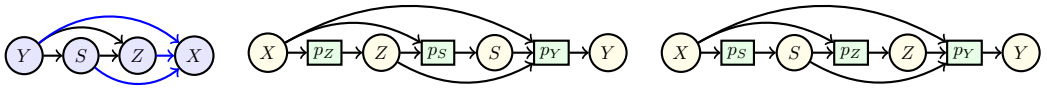


Figure 3: Bayes-Ball algorithm basic rules (Shachter, 2013). Conditional nodes are shaded.



(a) Bayesian Model (b) Z-First Inference (c) S-First Inference

Figure 4: Full-chain Bayesian graph and inference models for Z-first or S-first factorizations.

The Bayes-Ball algorithm (Shachter, 2013) facilitates an automatic pruning of redundant links in inference factor graphs through the analysis of conditional independency. Fig. 3 shows ten Bayes-Ball rules to identify conditional independency. Given a Bayesian graph, we can determine whether two disjoint sets of nodes are independent conditionally on other nodes through a graph separation criterion. Specifically, an undirected path is activated if a Bayes ball can travel along without encountering a stop symbol: $\longrightarrow\!\!|$ in Fig. 3. If there are no active paths between two nodes when some conditioning nodes are shaded, then those random variables are conditionally independent.

Graphical Models: We here focus on 4-node graphs. Let $p(y, s, z, x)$ denote the joint probability distribution underlying the datasets for the four random variables, i.e., Y, S, Z, and X. The chain rule can yield the following factorization for a generative model from Y to X (note that at most 4! factorization orders exist including useless ones such as reverse direction from X to Y):

$$p(y, s, z, x) = p(y)p(s|y)p(z|s, y)p(x|z, s, y), \tag{1}$$

which is visualized in Bayesian graph of Fig. 4(a). The probability conditioned on X can be factorized, e.g., as follows (among 3! different orders of inference factorization for four-node graphs):

$$p(y, s, z|x) = \begin{cases} p(z|x)p(s|z, x)p(y|s, z, x), & \text{Z-first-inference} \\ p(s|x)p(z|s, x)p(y|z, s, x), & \text{S-first-inference} \end{cases} \tag{2}$$

which are marginalized to obtain the likelihood: $p(y|x) = \mathbb{E}_{s,z|x}\left[p(y, s, z|x)\right]$. The above two scheduling strategies in (2) are illustrated in factor graph models as in Figs. 4(b) and (c), respectively.

The graphical models in Fig. 4 do not impose any assumption of potentially inherent independency in datasets and thus most generic. However, depending on underlying independency in datasets, we may be able to prune some edges in those graphs. For example, if the data only follows the simple Markov chain of $Y - X$, while being independent of S and Z, as shown in Fig. 5(a), all links except one between X and Y will be unreasonable in inference graphs of Figs. 4(b) and (c), that justifies the standard classifier model in Fig. 1(a). This implies that more complicated inference models such as A-CVAE can be unnecessarily redundant depending on the dataset. This motivates us to consider an extended AutoML framework which automatically explores the best pair of inference factor graph and corresponding Bayesian graph models matching dataset statistics besides the micro-scale hyperparameter tuning.

Methodology: AutoBayes begins with exploring any potential Bayesian graphs by cutting links of the full-chain graph in Fig. 4(a), imposing possible (conditional) independence. We then adopt the Bayes-Ball algorithm on each hypothetical Bayesian graph to examine conditional independence over different inference strategies, e.g., full-chain Z-/S-first inference graphs in Figs. 4(b)/(c). Applying Bayes-Ball justifies the reasonable pruning of the links in the full-chain inference graphs, and also the potential adversarial censoring when Z is independent of S. This process automatically constructs a connectivity of inference, generative, and adversary blocks with sound reasoning.

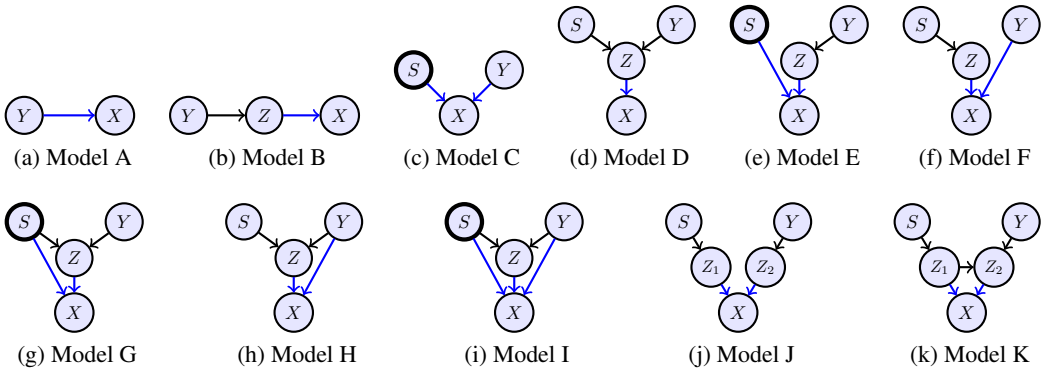


Figure 5: Example Bayesian graphs for data generative models under automatic exploration.

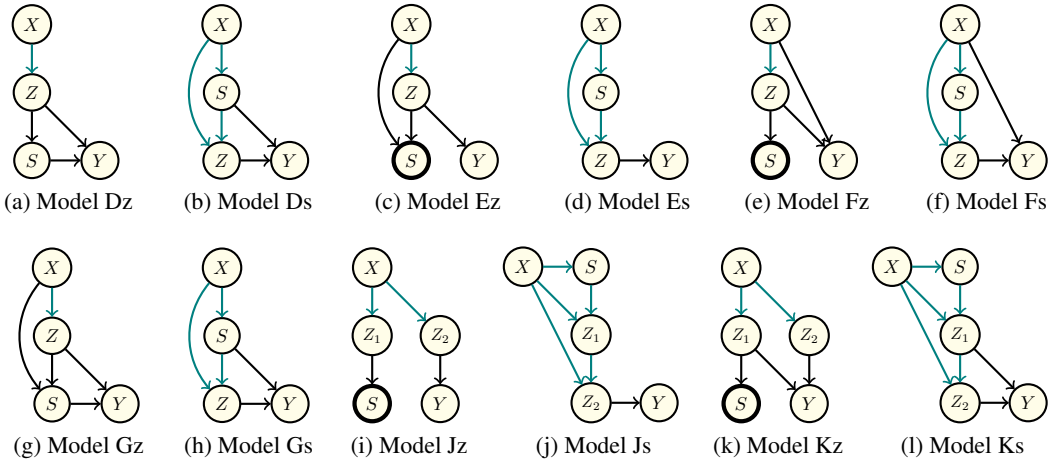


Figure 6: Z-first and S-first inference graph models relevant for generative models D–G, J, and K.

Consider an example case when the data adheres to the following factorization:

$$p(y, s, z, x) = p(y)p(s|\cancel{y})p(z|\cancel{s}, y)p(x|z, s, \cancel{y}), \tag{3}$$

where we explicitly indicate conditional independence by slash-cancellation from the full-chain case in (1). This corresponds to a Bayesian graphical model illustrated in Fig. 5(e). Applying the Bayes-Ball algorithm to the Bayesian graph yields the following conditional probability:

$$p(y, s, z|x) = p(z|x)p(s|z, x)p(y|z, \cancel{s}, \cancel{x}), \tag{4}$$

for the Z-first inference strategy in (2). The corresponding factor graph is then given in Fig. 6(c). Note that the Bayes-Ball also reveals that there is no marginal dependency between Z and S, which provides the reason to use adversarial censoring to suppress nuisance information S in the latent space Z. In consequence, by combining the Bayesian graph and factor graph, we automatically obtain A-CVAE model in Fig. 1(b). AutoBayes justifies A-CVAE under the assumption that the data follows the Bayesian model E in Fig. 5(e). As the true generative model is unknown, AutoBayes explores different Bayesian graphs like in Fig. 5 to search for the most relevant model. See Appendix A.2 for more detail descriptions for each Bayesian graph model to construct factor graphs like in Fig. 6. Also refer discussions of graphical models suitable for semi-supervised learning in Appendix A.4.

Training: Given a pair of generative graph and inference graph, the corresponding DNN structures will be trained. For example of the generative graph model K in Fig. 5(k), one relevant inference graph Kz in Fig. 6(k) will result in the overall network structure as shown in Fig. 7, where adversary network is attached as Z_2 is (conditionally) independent of S. This 5-node graph model justifies a

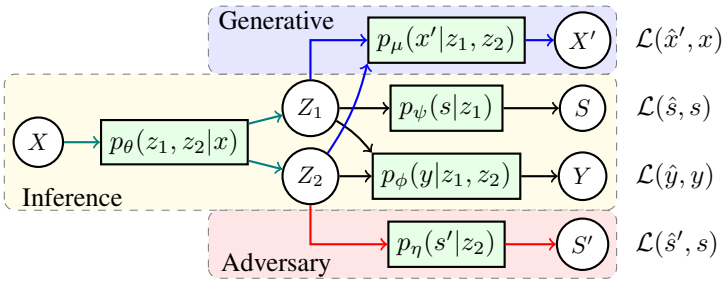


Figure 7: Overall network structure for pairing generative model K and inference model Kz.

recent work on partially disentanged A-CVAE by Han et al. (2020). Each factor block is realized by a DNN, e.g., parameterized by θ for $p_\theta(z_1, z_2|x)$, and all of the networks except for adversarial network are optimized to minimize corresponding loss functions including $\mathcal{L}(\hat{y}, y)$ as follows:

$$\min_{\theta,\psi,\mu} \max_{\eta} \mathbb{E}\big[\mathcal{L}(\hat{y}, y) + \lambda_s \mathcal{L}(\hat{s}, s) + \lambda_x \mathcal{L}(\hat{x}', x) + \lambda_z \mathbb{KL}(z_1, z_2 \| \mathcal{N}(\mathbf{0}, \mathbf{I})) - \lambda_a \mathcal{L}(\hat{s}', s)\big], \tag{5}$$

$$(z_1, z_2) = p_\theta(x), \quad \hat{y} = p_\phi(z_1, z_2), \quad \hat{s} = p_\psi(z_1), \quad \hat{x}' = p_\mu(z_1), \quad \hat{s}' = p_\eta(z_2), \tag{6}$$

where λ_* denotes a regularization coefficient, $\mathbb{KL}$ is the Kullback–Leibler divergence, and the adversary network $p_\eta(s'|z_2)$ is trained to minimize $\mathcal{L}(\hat{s}', s)$ in an alternating fashion (see the Adversarial Regularization paragraph below).

The training objective can be formally understood from a likelihood maximization perspective, in manner that can be seen as a generalization of the VAE Evidence Lower Bound (ELBO) concept. Specifically, it can be viewed as the maximization of a variational lower bound of the likelihood $p_\Phi(x, y, s)$ that is implicitly defined and parameterized by the networks, where Φ represents the collective parameters of the network modules (e.g., $\Phi = (\phi, \psi, \mu)$ in the example of equation 5) that specify the generative model $p_\Phi(x, y, s|z)$, which implies the likelihood $p_\Phi(x, y, s)$, as given by

$$p_\Phi(x, y, s) = \int p_\Phi(x, y, s|z) p(z) \, dz.$$

However, since this expression is generally intractable, we introduce $q_\theta(z|x, y, s)$ as a variational approximation of the posterior $p_\Phi(z|x, y, s)$ implied by the generative model:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \log p_\Phi(x_i, y_i, s_i) &= \frac{1}{n}\sum_{i=1}^{n}\left[\log p_\Phi(x_i, y_i, s_i|z_i) - \log \frac{q_\theta(z_i|x_i, y_i, s_i)}{p(z_i)} + \log \frac{q_\theta(z_i|x_i, y_i, s_i)}{p_\Phi(z_i|x_i, y_i, s_i)}\right] \\
&\approx \frac{1}{n}\sum_{i=1}^{n} [\log p_\Phi(x_i, y_i, s_i|z_i)] - \mathbb{KL}(q_\theta(z|x, y, s) \| p(z)) + \mathbb{KL}(q_\theta(z|x, y, s) \| p_\Phi(z|x, y, s)) \\
&\geq \frac{1}{n}\sum_{i=1}^{n} [\log p_\Phi(x_i, y_i, s_i|z_i)] - \mathbb{KL}(q_\theta(z|x, y, s) \| p(z)),
\end{aligned} \tag{7}$$

where the samples $z_i \sim q_\theta(z|x_i, y_i, s_i)$ are drawn for each training tuple (x_i, y_i, s_i), and the final inequality follows from the non-negativity of KL-divergence.

Ultimately, the minimization of our training loss function corresponds to the maximization of the lower bound in (7), which corresponds to maximizing the likelihood of our implicit generative model, while also optimizing the variational posterior $q_\theta(z|x, y, s)$ toward the actual posterior for the latent representation $p_\Phi(z|x, y, s)$, since the gap in the bound is given by $\mathbb{KL}(q_\theta(z|x, y, s) \| p_\Phi(z|x, y, s))$. Further factoring of $\log p_\Phi(x, y, s|z)$ yields the multiple loss-terms and network modules.

Adversarial Regularization: We can utilize adversarial censoring when Z and S should be marginally independent, e.g., such as in Fig. 1(b) and Fig. 7, in order to reinforce the learning of a representation Z that is disentangled from the nuisance variations S. This is accomplished by introducing an adversarial network that aims to maximize a parameterized approximation $q(s|z)$ of the likelihood $p(s|z)$, while this likelihood is also incorporated into the loss for the other modules with

Table 1: Parameters of Public Dataset Under Investigation

Dataset	Modality	Dimension	Nuisance ($\lvert S \rvert$)	Classes ($\lvert Y \rvert$)	Samples	Reference
QMNIST	Image	28×28	836	10	70,000	Yadav & Bottou (2019)
Stress	Temperature etc.	7×1	20	4	24,000	Birjandtalab et al. (2016)
RSVP	EEG	16×128	10	4	41,400	Orhan et al. (2012)
MI	EEG	64×480	106	4	9,540	Goldberger et al. (2000)
ErrP	EEG	56×250	27	2	9,180	Margaux et al. (2012)
Faces Basic	ECoG	31×400	14	2	4,100	Miller et al. (2015; 2016)
Faces Noisy	ECoG	39×400	7	2	2,100	Miller et al. (2015; 2017)
ASL	EMG	16×100	5	33	9,900	Günay et al. (2019)

a negative weight. The adversarial network, by maximizing the log likelihood $\log q(s|z)$, essentially maximizes a lower-bound of the mutual information $\mathbb{I}(S; Z)$, and hence the main network is regularized with the additional term that corresponds to minimizing this estimate of mutual information. This follows since the log-likelihood maximized by the adversarial network is given by

$$\mathbb{E}[\log q(s|z)] = \mathbb{I}(S; Z) - \mathbb{H}(S) - \mathbb{KL}\big(p(s|z) \| q(s|z)\big), \tag{8}$$

where the entropy $\mathbb{H}(S)$ is constant.

Ensemble Learning: We further introduce ensemble methods to make best use of all Bayesian graph models explored by the AutoBayes framework without wasting lower-performance models. Ensemble stacked generalization works by stacking the predictions of the base learners in a higher level learning space, where a meta learner corrects the predictions of base learners (Wolpert, 1992). Subsequent to training base learners, we assemble the posterior probability vectors of all base learners together to improve the prediction. We compare the predictive performance of a logistic regression (LR) and a shallow multi-layer perceptron (MLP) as an ensemble meta learner to aggregate all inference models. See Appendix A.5 for more detailed description of the stacked generalization.

4 Experimental Evaluation

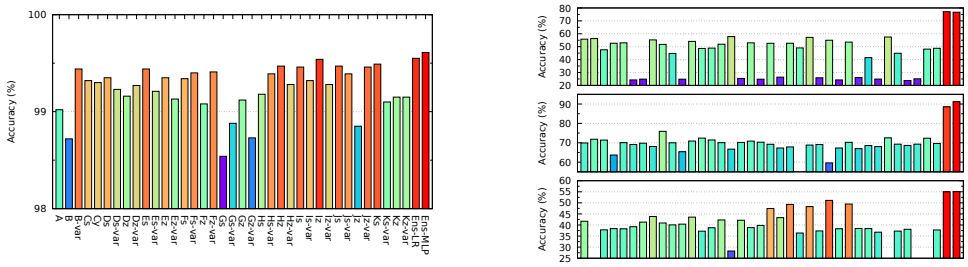
Datasets: We experimentally demonstrate the performance of AutoBayes for publicly available datasets as listed in Table 1. Note that they cover a wide variety of data size, dimensionality, subject scale, and class levels as well as sensor modalities including image, EEG, EMG, and electrocorticography (ECoG). See more detailed information of each dataset in Appendix A.6.

Model Implementation: All models were trained with a minibatch size of 32 and using the Adam optimizer with an initial learning rate of 0.001. The learning rate is halved whenever the validation loss plateaus. A compact convolutional neural network (CNN) with 4 layers is employed as an encoder network $\mathcal{E}$ to extract features from $C \times T$ data. Each convolution is followed by batch normalization (BN) and rectified linear unit (ReLU) activation. The AutoBayes chooses either a deterministic latent encoder or variational latent encoder under Gaussian prior. The original data is reconstructed by a decoder network $\mathcal{D}$ that applies transposed convolutions. See Appendix A.7 for more details.

Results: Fig. 8(a) shows the performance of QMNIST across 39 different inference models explored in AutoBayes including 2 ensemble models. Over 37 base models, some outperforms the standard classifier model A, whereas the rest of the models underperform. We observe a large gap of 1.0% between the best and worst models with a standard deviation of 0.23% across all Bayesian models. This indicates that we may have a potential risk that one particular model may lose up to 1.0% accuracy if we do not explore different models.

Similar behaviors with a huge deviation can be seen for different datasets as shown in Fig. 8(b). It was shown that the best inference strategy highly depends on datasets. Specifically, the best model at one dataset does not perform best for different datasets. This suggests that we must consider different inference strategies for each target dataset and our AutoBayes provides such an adaptive framework across datasets. More results are found in Appendix A.8.

Remarkably, the ensemble of base learners further enhances the performance regardless of the choice from LR or MLP as the meta learner, as illustrated in Fig. 2 across all the datasets. For some low-



(a) QMNIST (±0.23%)

(b) MI (±13.1%), ErrP (±7.0%), ASL (±12.0%)

Figure 8: Task classification accuracy across different graphical models (with standard deviation).

Table 2: Task classification performance of AutoBayes compared to state-of-the-art.

Method	QMNIST	Stress	RSVP	MI	ErrP	Faces Basic	Faces Noisy	ASL
Ensemble AutoBayes (Meta-MLP)	99.61	**98.98**	**99.99**	76.71	**91.21**	**78.36**	**89.71**	**55.06**
Ensemble AutoBayes (Meta-LR)	99.55	98.96	99.98	**77.14**	88.54	75.68	88.40	54.98
Best of AutoBayes	99.54	95.35	93.42	57.83	75.91	67.31	76.58	51.12
State-of-the-art (SOTA)	**99.84**	85.30	71.60	63.8	48.80	—	—	—

performing datasets such as ErrP, MI and Faces (Noisy), ensemble learning significantly improves the accuracy by 15.3%, 19.3% and 13.2% at the expense of more storage and computational resources.

We benchmark the performance of AutoBayes against the benchmark competitor models from (Byerly et al., 2020; Han et al., 2020; Özdenizci et al., 2019c;b;a) in Table 2. It can be seen that AutoBayes outperforms the state-of-the-art in all datasets except QMNIST. Consequently, we can see a great advantage of AutoBayes with exploring different graphical models. Even for QMNIST, AutoBayes meta-MLP model, achieving 99.61% accuracy, ranks 17 in the published leaderboard. Note that performing better than 99.84% is nearly impossible, since some numbers are illegible or mislabeled. Also note that we have not specifically designed AutoBayes architecture for image classification but for spatio-temporal signal applications and micro-scale hyper-parameters were not fully optimized yet.

AutoBayes can be readily integrated with AutoML to optimize any hyperparameters of individual DNN blocks. Nevertheless, as our primary objective was to show a proof-of-concept benefit from solely graphical model exploration of AutoBayes, we leave more rigorous analysis to optimize DNN parameters such as network depths, widths, activation, augmentation, etc. as a future work.

5 Conclusion and Future Work

We proposed a new concept called AutoBayes which explores various different Bayesian graph models to facilitate searching for the best inference strategy, suited for nuisance-robust deep learning. With the Bayes-Ball algorithm, our method can automatically construct reasonable link connections among classifier, encoder, decoder, nuisance estimator and adversary DNN blocks. As a proof-of-concept analysis, we demonstrated the benefit of AutoBayes for various public datasets. We observed a huge performance gap between the best and worst graph models, implying that the use of one deterministic model without graph exploration can potentially suffer a poor classification result. In addition, the best model for one dataset does not always perform best for different data, which encourages us to use AutoBayes for adaptive model generation given target datasets. We further improved the performance approaching the state-of-the-art accuracy by exploiting multiple graphical models explored in AutoBayes through the use of ensemble stacking. We are extending the AutoBayes framework to integrate AutoML to optimize hyperparameters of each DNN block. How to handle the exponentially growing search space of possible Bayesian graphs along with the number of random variables remains a challenging future work.

REFERENCES

Anubhav Ashok, Nicholas Rhinehart, Fares Beainy, and Kris M Kitani. N2N learning: Network to network compression via policy gradient reinforcement learning. *arXiv preprint arXiv:1709.06030*, 2017.

Manfredo Atzori, Matteo Cognolato, and Henning Müller. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in neurorobotics*, 10:9, 2016.

Justin Bayer, Daan Wierstra, Julian Togelius, and Jürgen Schmidhuber. Evolving memory cell structures for sequence learning. In *International Conference on Artificial Neural Networks*, pp. 755–764. Springer, 2009.

Javad Birjandtalab, Diana Cogan, Maziyar Baran Pouyan, and Mehrdad Nourani. A non-EEG biosignals dataset for assessment and visualization of neurological status. In *2016 IEEE International Workshop on Signal Processing Systems (SiPS)*, pp. 110–114. IEEE, 2016.

Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Smash: one-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344*, 2017.

Adam Byerly, Tatiana Kalganova, and Ian Dear. A branching and merging convolutional network with homogeneous filter capsules. *arXiv preprint arXiv:2001.09136*, 2020.

Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Reinforcement learning for architecture search by network transformation. *arXiv preprint arXiv:1707.04873*, 2017.

Christoforos Christoforou, Robert M Haralick, Paul Sajda, and Lucas C Parra. The bilinear brain: towards subject-invariant analysis. In *2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pp. 1–6. IEEE, 2010.

Antonia Creswell, Anil A Bharath, and Biswa Sengupta. Conditional autoencoders with adversarial information factorization. *arXiv preprint arXiv:1711.05175*, 2017.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 113–123, 2019.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

Oliver Faust, Yuki Hagiwara, Tan Jen Hong, Oh Shu Lih, and U Rajendra Acharya. Deep learning for healthcare applications based on physiological signals: A review. *Computer methods and programs in biomedicine*, 161:1–13, 2018.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Sezen Yağmur Günay, Mathew Yarossi, Dana H Brooks, Eugene Tunik, and Deniz Erdoğmuş. Transfer learning using low-dimensional subspaces for EMG-based classification of hand posture. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 1097–1100. IEEE, 2019.

Mo Han, Ozan Özdenizci, Ye Wang, Toshiaki Koike-Akino, and Deniz Erdogmus. Disentangled adversarial autoencoder for subject-invariant physiological feature extraction. *IEEE Signal Processing Letters*, pp. 1565–1569, 2020.

Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. AMC: AutoML for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 784–800, 2018.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pp. 2342–2350, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pp. 5967–5976, 2017.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Perrin Margaux, Maby Emmanuel, Daligault Sébastien, Bertrand Olivier, and Mattout Jérémie. Objective and subjective evaluation of online error correction during P300-based spelling. *Advances in Human-Computer Interaction*, 2012, 2012.

Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, et al. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pp. 293–312. Elsevier, 2019.

Kai J Miller, Dora Hermes, Nathan Witthoft, Rajesh PN Rao, and Jeffrey G Ojemann. The physiology of perception in human temporal lobe is specialized for contextual novelty. *Journal of neurophysiology*, 114(1):256–263, 2015.

Kai J Miller, Gerwin Schalk, Dora Hermes, Jeffrey G Ojemann, and Rajesh PN Rao. Spontaneous decoding of the timing and content of human object perception from cortical surface recordings reveals complementary information in the event-related potential and broadband spectral change. *PLoS computational biology*, 12(1), 2016.

Kai J Miller, Dora Hermes, Franco Pestilli, Gagan S Wig, and Jeffrey G Ojemann. Face percept formation in human ventral temporal cortex. *Journal of neurophysiology*, 118(5):2614–2627, 2017.

Marvin Minsky and Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

Umut Orhan, Kenneth E Hild, Deniz Erdoğmuş, Brian Roark, Barry Oken, and Melanie Fried-Oken. RSVP keyboard: An EEG based typing interface. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 645–648. IEEE, 2012.

Ozan Özdenizci, Barry Oken, Tab Memmott, Melanie Fried-Oken, and Deniz Erdogmus. Adversarial feature learning in brain interfacing: An experimental study on eliminating drowsiness effects. *arXiv preprint arXiv:1907.09540*, 2019a.

Ozan Özdenizci, Ye Wang, Toshiaki Koike-Akino, and Deniz Erdoğmuş. Transfer learning in brain-computer interfaces with adversarial variational autoencoders. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 207–210. IEEE, 2019b.

Ozan Özdenizci, Ye Wang, Toshiaki Koike-Akino, and Deniz Erdoğmuş. Adversarial deep learning in eeg biometrics. *IEEE Signal Processing Letters*, 26(5):710–714, 2019c.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. SpecAugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2902–2911. JMLR. org, 2017.

Esteban Real, Chen Liang, David R So, and Quoc V Le. AutoML-Zero: Evolving machine learning algorithms from scratch. *arXiv preprint arXiv:2003.03384*, 2020.

Ross D Shachter. Bayes-Ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *arXiv preprint arXiv:1301.7412*, 2013.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pp. 3483–3491, 2015.

Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

Chhavi Yadav and Léon Bottou. Cold case: The lost MNIST digits. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.