

APPENDICES

A.1 RELATED WORK

We note that this paper relates to some existing literature as follows.

- **AutoML:** Searching DNN models with hyperparameter optimization has been intensively investigated in a framework called 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). The automated methods include architecture search (Zoph et al., 2018; Real et al., 2017; He et al., 2018; Real et al., 2020), learning rule design (Bayer et al., 2009; Jozefowicz et al., 2015), and augmentation exploration (Cubuk et al., 2019; Park et al., 2019). Most work used either evolutionary optimization or reinforcement learning framework to adjust hyperparameters or to construct network architecture from pre-selected building blocks. The recent AutoML-Zero (Real et al., 2020) considers an extension to preclude human knowledge and insights for fully automated designs from scratch.
- **Variational Bayesian Inference:** The VAE (Kingma & Welling, 2013) introduced variational Bayesian inference methods, incorporating autoassociative architectures, where generative and inference models can be learned jointly. This method was extended with the CVAE (Sohn et al., 2015), which introduces a conditioning variable that could be used to represent nuisance variations, and a regularized VAE in (Louizos et al., 2015), which considers disentangling the nuisance variable from the latent representation.
- **Adversarial Training:** The concept of adversarial networks was introduced with Generative Adversarial Networks (GAN) (Goodfellow et al., 2014), and has been adopted into myriad applications. The simultaneously discovered Adversarially Learned Inference (ALI) (Dumoulin et al., 2016) and Bidirectional GAN (BiGAN) (Donahue et al., 2016) propose an adversarial approach toward training an autoencoder. Adversarial training has also been combined with VAE to regularize and/or disentangle the latent representations (Makhzani et al., 2015; Lample et al., 2017; Creswell et al., 2017).

A.2 BAYESIAN GRAPH AND INFERENCE MODELS

Given measurement data, we never know the true joint probability beforehand, and therefore we shall assume one of several possible generative models. AutoBayes aims to explore such potential graph models to match the measurement distributions. As the maximum possible number of graphical models is huge even for a four-node case involving Y , S , Z and X , we restrict our focus to a few meaningful graphs-of-interest shown in Fig. 5. Each Bayesian graph corresponds to the following assumption of the joint probability factorization ($p(x|\dots)$ term specifies a generative model of X):

$$p(y, s, z, x) = \begin{cases} p(y)p(s|y)p(z|\cancel{s}, y)p(x|\cancel{z}, \cancel{s}, y), & \text{Model-A} \\ p(y)p(s|y)p(z|\cancel{s}, y)p(x|z, \cancel{s}, y), & \text{Model-B} \\ p(y)p(s|y)p(z|\cancel{s}, y)p(x|\cancel{z}, s, y), & \text{Model-C} \\ p(y)p(s|y)p(z|s, y)p(x|\cancel{z}, \cancel{s}, y), & \text{Model-D} \\ p(y)p(s|y)p(z|\cancel{s}, y)p(x|z, s, \cancel{y}), & \text{Model-E} \\ p(y)p(s|y)p(z|s, y)p(x|\cancel{z}, \cancel{s}, y), & \text{Model-F} \\ p(y)p(s|y)p(z|s, y)p(x|z, s, y), & \text{Model-G} \\ p(y)p(s|y)p(z|s, y)p(x|z, \cancel{s}, y), & \text{Model-H} \\ p(y)p(s|y)p(z|s, y)p(x|z, s, y), & \text{Model-I} \\ p(y)p(s|y)p(z_1|s, y)p(z_2|\cancel{z_1}, \cancel{s}, y)p(x|z_2, z_1, \cancel{s}, y), & \text{Model-J} \\ p(y)p(s|y)p(z_1|s, y)p(z_2|z_1, \cancel{s}, y)p(x|z_2, z_1, \cancel{s}, y), & \text{Model-K} \end{cases} \quad (9)$$

where we explicitly indicate independence by slash-cancelled factors from the full-chain case in equation 1. Depending on the assumed Bayesian graph, the relevant inference strategy will vary as some variables may be conditionally independent, which enables pruning links in the inference factor graphs. As shown in Fig. 6, the reasonable inference graph model can be automatically generated by the Bayes-Ball algorithm (Shachter, 2013) on each Bayesian graph hypothesis inherent in datasets. Specifically, the conditional probability $p(y, s, z|x)$ can be obtained for each model as below.

Bayesian Graph Model A (Direct Markov): The simplest model between X and Y would be single Markov chain without any dependency of S and Z , shown in Bayesian graph of Fig. 5(a). This model puts an assumption that the data are nuisance-invariant. For this case, there is no reason to employ complicated inference models such as A-CVAE since most factors will be independent as $p(y, s, z|x) = p(z|x)p(s|z, x)p(y|s, z, x)$. We hence should use a standard classification method, as in Fig. 1(a), to infer Y given X , based on the inference model $p(y|x)$ without involving S and Z .

Bayesian Graph Model B (Markov Latent): Assuming a latent Z can work in a Markov chain of $Y - Z - X$ shown in Fig. 5(b), we obtain a simple inference model: $p(y, s, z|x) = p(z|x)p(s|z, x)p(y|s, z, x)$. Note that this model assumes independence between Z and S , and thus adversarial censoring (Makhzani et al., 2015; Creswell et al., 2017; Lample et al., 2017) can make it more robust against nuisance.

Bayesian Graph Model C (Subject-Dependent): We may model the case when the data X directly depends on subject S and task Y , shown in Fig. 5(c). For this case, we may consider the corresponding inference models due to the Bayes-Ball:

$$p(y, s, z|x) = \begin{cases} p(s|x)p(z|s, x)p(y|s, z, x), & \text{Model-Cs} \\ p(y|x)p(s|y, x)p(z|s, y, x), & \text{Model-Cy} \end{cases} \quad (10)$$

Note that this model does not depend on Z , and thus Z -first inference strategy reduces to S -first model. As a reference, we here consider additional Y -first inference strategy to evaluate the difference.

Bayesian Graph Model D (Latent Summary): Another graphical model is shown in Fig. 5(d), where a latent space bridges all other random variables. Bayes-Ball yields the following models:

$$p(y, s, z|x) = \begin{cases} p(z|x)p(s|z, x)p(y|s, z, x), & \text{Model-Dz} \\ p(s|x)p(z|s, x)p(y|z, s, x), & \text{Model-Ds} \end{cases} \quad (11)$$

whose graphical models are depicted in Figs. 6(a) and (b), respectively.

Bayesian Graph Model E (Task-Summary Latent): Another graphical model involving latent variables is shown in Fig. 5(e), where a latent space only summarizes Y . Bayes-Ball yields the following inference models:

$$p(y, s, z|x) = \begin{cases} p(z|x)p(s|z, x)p(y|z, s, x), & \text{Model-Ez} \\ p(s|x)p(z|s, x)p(y|s, z, x), & \text{Model-Es} \end{cases} \quad (12)$$

which are illustrated in Figs. 6(c) and (d). Note that the generative model E has 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 addition, because the generative model of X is dependent on both Z and S , it is justified to employ the A-CVAE classifier shown in Fig. 1(b).

Bayesian Graph Model F (Subject-Summary Latent): Consider Fig. 5(f), where a latent variable summarizes subject information S . The Bayes-Ball provides the inference graphs shown in Figs. 6(e) and (f), which respectively correspond to:

$$p(y, s, z|x) = \begin{cases} p(z|x)p(s|z, x)p(y|s, z, x), & \text{Model-Fz} \\ p(s|x)p(z|s, x)p(y|x, s, z), & \text{Model-Fs} \end{cases} \quad (13)$$

Bayesian Graph Model G: Letting the joint distribution follow the model G in Fig. 5(g), we obtain the following inference models via the Bayes-Ball:

$$p(y, s, z|x) = \begin{cases} p(z|x)p(s|z, x)p(y|s, z, x), & \text{Model-Gz} \\ p(s|x)p(z|s, x)p(y|z, s, x), & \text{Model-Gs} \end{cases} \quad (14)$$

whose graphical models are described in Figs. 6(g) and (h). Note that the inference model Gs in Fig. 6(h) is identical to the inference model Ds in Fig. 6(b). Although the inference graphs Gs and Ds are identical, the generative model of X is different as shown in Figs. 5(g) and (d). Specifically, VAE decoder for the model G should feed S along with variational latent space Z , and thus using CVAE is justified for the model G but D. This difference of the generative models can potentially make a different impact on the performance of inference despite the inference graph alone is identical.

Bayesian Graph Models H and I: Both the generative models H and I shown in Figs. 5(h) and (i) have the fully-connected inference strategies as given in (2), whose graphs are shown in Figs. 4(b) and (c), respectively, since no useful conditional independency can be found with the Bayes-Ball. Analogous to the relation of models Ds and Gs, the inference graph can be identical for Bayesian graphs H and I, whereas the generative model of X is different as shown in Figs. 5(h) and (i).

Bayesian Graph Model J (Disentangled Latent): We can also consider multiple latent vectors to generalize the Bayesian graph with more vertices. We here focus on two such examples of graph models with two-latent spaces as shown in Figs. 5(j) and (k). Those models are identical class of the model D, except that a single latent Z is disentangled into two parts Z_1 and Z_2 , respectively associated with S and Y . Given the Bayesian graph of Fig. 5(j), the Bayes-Ball yields some inference strategies including the following two models:

$$p(y, s, z_1, z_2|x) = \begin{cases} p(z_1, z_2|x)p(s|z_1, \cancel{z_2}, \cancel{x})p(y|\cancel{s}, \cancel{z_1}, z_2, \cancel{x}), & \text{Model-Jz} \\ p(s|x)p(z_1|s, x)p(z_2|\cancel{s}, z_1, x)p(y|\cancel{s}, \cancel{z_1}, z_2, \cancel{x}), & \text{Model-Js} \end{cases} \quad (15)$$

which are shown in Figs. 6(i) and (j). Note that Z_2 is marginally independent of the nuisance variable S , which encourages the use of adversarial training to be robust against subject/session variations.

Bayesian Graph Model K (Conditionally Disentangled Latent): Another modified model in Fig. 5(k) linking Z_1 and Z_2 yields the following inference models:

$$p(y, s, z_1, z_2|x) = \begin{cases} p(z_1, z_2|x)p(s|z_1, \cancel{z_2}, \cancel{x})p(y|\cancel{s}, z_1, z_2, \cancel{x}), & \text{Model-Kz} \\ p(s|x)p(z_1|s, x)p(z_2|\cancel{s}, z_1, x)p(y|\cancel{s}, z_1, z_2, \cancel{x}), & \text{Model-Ks} \end{cases} \quad (16)$$

as shown in Figs. 6(k) and (l). The major difference from the model J lies in the fact that the inference graph should use Z_1 along with Z_2 to infer Y .

A.3 BACKGROUND ON VARIATIONAL BAYESIAN INFERENCE

Variational AE AutoBayes may automatically construct autoencoder architecture when latent variables are involved, e.g., for the model E in Fig. 5(e). For this case, Z represents a stochastic node to marginalize out for X reconstruction and Y inference, and hence VAE will be required. In contrast to vanilla autoencoders, VAE uses variational inference by assuming a marginal distribution for latent $p(z)$. In variational approach, we reparameterize Z from a prior distribution such as the normal distribution to marginalize. Depending on the Bayesian graph models, we can also consider reparameterizing semi-supervision on S (i.e., incorporating a reconstruction loss for S) as a conditioning variable. Conditioning on Y and/or S should depend on consistency with the graphical model assumptions. Since VAE is a special case of CVAE, we will go into further detail about the more general CVAE below.

Conditional VAE When X is directly dependent on S or Y along with Z in the Bayesian graph, the AutoBayes gives rise the CVAE architecture, e.g., for the models E/F/G/H/I in Fig. 5. For those generative models, the decoder DNN needs to feed S or Y as a conditioning parameter. Even for other Bayesian graphs, the S -first inference strategy will require conditional encoder in CVAE, e.g., the models Ds/Es/Fs/Gs/Js/Ks in Fig. 6, where latent Z depends on S .

Consider the case when S plays as the conditioning variable in a data model with the factorization:

$$p(s, x, z) = p(s)p(z)p(x|s, z), \quad (17)$$

where we directly parameterize $p(x|s, z)$, set $p(z)$ to something simple (e.g., isotropic Gaussian), and leave $p(s)$ arbitrary (since it will not be directly used). The CVAE is trained according to maximizing the likelihood of data tuples (s, x) with respect to $p(x|s)$, which is given by

$$p(x|s) = \int p(x|s, z)p(z) dz, \quad (18)$$

which is intractable to compute exactly given the potential complexity of the parameterization of $p(x|s, z)$. While it could be possible to approximate the integration with sampling of Z , the crux of the VAE approach is to utilize a variational lower-bound of the likelihood that involves a variational

approximation of the posterior $p(z|s, x)$ implied by the generative model. With $q(z|s, x)$ representing the variational approximation of the posterior, the Evidence Lower-Bound (ELBO) is given by

$$\log p(x|s) \geq \mathbb{E}_{z \sim q(z|s, x)} [\log p(x|s, z)] - \mathbb{KL}(q(z|s, x) \| p(z)). \quad (19)$$

The parameterization of the variational posterior $q(z|s, x)$ may also be decomposed into parameterized components, e.g., $q(z|s, x) = q(s|x)q(z|s, x)$ such as in the S -first models shown in Fig. 6. Such decomposition also enables the possibility of semi-supervised training, which can be convenient when some of the variables, such as the nuisances variations, are not always labeled. For data tuples that include s , the likelihood $q(s|x)$ can also be directly optimized, and the given value for s is used as an input to the computation of $q(z|s, x)$. However, for tuples where s is missing, the component $q(s|x)$ can be used to generate an estimate of s to be input to $q(z|s, x)$. We further discuss semi-supervised learning and the sampling methods for categorical nuisance variables in Appendix A.4 below.

A.4 SEMI-SUPERVISED LEARNING: CATEGORICAL SAMPLING

Graphical Models for Semi-Supervised Learning Nuisance values S such as subject ID or session ID may not be always available for typical physiological datasets, in particular for the testing phase of an HMI system deployment with new users, requiring semi-supervised methods. We note that some graphical models are well-suited for such semi-supervised training. For example, among the Bayesian graph models in Fig. 5, the models C/E/G/I require the nuisance S to reproduce X . If no ground-truth labels of S are available, we need to marginalize S across all possible categories for the decoder DNN \mathcal{D} . Even for other Bayesian graphs, the corresponding inference factor graphs in Fig. 6 may not be convenient for the semi-supervised settings. Specifically, for models Ez/Fz/Jz/Kz have an inference of S at the end node, whereas the other inference models use inferred S for subsequent inference of other parameters. If S is missing or unknown as a semi-supervised setting, those inference graphs having S in a middle node are inconvenient as we need sampling over all possible nuisance categories. For instance, the model Kz shown in Fig. 7 does not need S marginalization, and thus readily applicable to semi-supervised datasets.

Variational Categorical Reparameterization In order to deal with the issue of categorical sampling, we can use the Gumbel-Softmax reparameterization trick (Jang et al., 2016), which enables differentiable approximation of one-hot encoding. Let $[\pi_1, \pi_2, \dots, \pi_{|S|}]$ denote a target probability mass function for the categorical variable S . Let $g_1, g_2, \dots, g_{|S|}$ be independent and identically distributed samples drawn from the Gumbel distribution $\text{Gumbel}(0, 1)$.¹ Then, generate an $|S|$ -dimensional vector $\hat{s} = [\hat{s}_1, \hat{s}_2, \dots, \hat{s}_{|S|}]$ according to

$$\hat{s}_k = \frac{\exp((\log(\pi_k) + g_k)/\tau)}{\sum_{i=1}^{|S|} \exp((\log(\pi_i) + g_i)/\tau)}, \quad (20)$$

where $\tau > 0$ is a softmax temperature. As the softmax temperature τ approaches 0, samples from the Gumbel-Softmax distribution become one-hot and the distribution becomes identical to the target categorical distribution. The temperature τ is usually decreased across training epochs as an annealing technique, e.g., with exponential decaying.

A.5 ENSEMBLE LEARNING: STACKED GENERALIZATION

To achieve higher predictive performance, we construct ensembles from the output posterior class probabilities of all graphical models. Let $\mathcal{D}_0 = \{(x_n, y_n, s_n) | n = 1 : N\}$ denote a data set, where x_n is a data instance, y_n is the task label, s_n is the nuisance (subject) label and N is the number of samples in the dataset. We randomly split the data into training set $\mathcal{D}_{\text{train}}$ and validation set $\mathcal{D}_{\text{test}}$. Given 37 graphical models, which we call base learners, we induce a decision algorithm \mathcal{M}_k , for $k = 1, \dots, 37$ by invoking the k th graphical model on the data in $\mathcal{D}_{\text{train}}$. For each x_n in $\mathcal{D}_{\text{train}}$, graphical model \mathcal{M}_k generates a class probability vector for task and nuisance label prediction. Let $P_{ky}(x_n) = \{P(y_1|x_n), \dots, P(y_i|x_n), \dots, P(y_{N_y}|x_n)\}$ denote the posterior probability distribution over N_y task labels and $P_{ks}(x_n) = \{P(s_1|x_n), \dots, P(s_i|x_n), \dots, P(s_{N_s}|x_n)\}$ denote the posterior

¹The $\text{Gumbel}(0, 1)$ distribution can be sampled using inverse transform sampling by drawing $u \sim \text{Uniform}(0, 1)$ and computing $g = -\log(-\log(u))$.

probability distribution over N_s nuisance labels produced by model \mathcal{M}_k given data instance x_n . Ensemble generalization works by stacking the predictions of the base learners in a higher level learning space, where meta learner, denoted as $\tilde{\mathcal{M}}_k$, corrects the predictions of base learners (Wolpert, 1992). Subsequent to training base learners, we assemble the posterior probability vectors of all base learners together: $P_y(x_n) = \{P_{ky}(x_n)\}$ and $P_s(x_n) = \{P_{ks}(x_n)\}$, where $k = 1 : 37$. $\tilde{\mathcal{M}}_k$ is trained using the predictions from all base learners as input attributes: $\mathcal{D}_{\text{train}}^{\text{in}} = \{(P_y(x_n), P_s(x_n))\}$ and correct labels as output: $\mathcal{D}_{\text{train}}^{\text{out}} = \{(y_n, s_n)\}$, where $n = 1 : N_{\text{train}}$. Hold-out $\mathcal{D}_{\text{test}}$ is used to measure the classification performance of both base and meta learners. To make best use of the base learners, we compare the predictive performance of a LR model and a shallow MLP as a meta learner in Table 2.

A.6 DATASETS DESCRIPTION

We used publicly available physiological datasets as well as a benchmark MNIST as follows. The parameters of datasets are also summarized in Table 1.

- **QMNIIST:** A hand-written digit image MNIST with extended label information including a writer ID number (Yadav & Bottou, 2019).² There are $|S| = 539$ writers for classifying $|Y| = 10$ digits from grayscale 28×28 pixel images over 60,000 training samples. Additional 297 writers provide 10,000 test samples.
- **Stress:** A physiological dataset considering neurological stress level (Birjandtalab et al., 2016).³ It consists of multi-modal biosignals for $|Y| = 4$ discrete stress states from $|S| = 20$ healthy subjects, including physical/cognitive/emotional stresses as well as relaxation. The data were collected by $C = 7$ sensors, i.e., electrodermal activity, temperature, three-dimensional acceleration, heart rate, and arterial oxygen level. For each stress status, a corresponding task of 5 minutes long (i.e., $T = 300$ time samples with 1 Hz down-sampling) was assigned to subjects for a total of 4 trials.
- **RSVP:** An EEG-based typing interface using rapid serial visual presentation (RSVP) paradigm (Orhan et al., 2012).⁴ $|S| = 10$ healthy subjects participated in the experiments at three sessions performed on different days. The dataset consists of 41,400 epochs of $C = 16$ channel EEG data for $T = 128$ samples, which were collected by g.USBamp biosignal amplifier with active electrodes during RSVP keyboard operations. $|Y| = 4$ labels for emotion elicitation, resting-state, or motor imagery/execution task.
- **MI:** The PhysioNet EEG Motor Imagery (MI) dataset (Goldberger et al., 2000).⁵ Excluding irregular timestamp, the dataset consists of $|S| = 106$ subjects' EEG data. During the experiments, subjects were instructed to perform cue-based motor execution/imagery tasks while $C = 64$ channels were recorded at a sampling rate of 160 Hz. Focusing on motor imagery tasks, we use the EEG data for three seconds of post-cue interval data (i.e., $T = 480$ time samples). The subject performed $|Y| = 4$ -class tasks; either right hand motor imagery, left hand motor imagery, both hands motor imagery, or both feet motor imagery. This resulted in a total of 90 trials per subject.
- **ErrP:** An error-related potential (ErrP) of front-central EEG dataset (Margaux et al., 2012).⁶ The dataset consists of EEG data recorded from $|S| = 16$ healthy subjects participating in an offline P300 spelling task, where visual feedback of the inferred letter is provided to the user at the end of each trial for 1.3 seconds to monitor evoked brain responses for erroneous decisions made by the system. EEG data were recorded from $C = 56$ channels for epoched 1.25 seconds at a sampling rate of 200 Hz (i.e., $T = 250$). Across five recording sessions, each subject performed a total of 340 trials. Since it was an offline copy spell task, binary $|Y| = 2$ labels were provided as erroneous or correct feedback.

²QMNIIST dataset: <https://github.com/facebookresearch/qmnist>

³Stress dataset: <https://physionet.org/content/noneeg/1.0.0/>

⁴RSVP dataset: <http://hdl.handle.net/2047/D20294523>

⁵MI dataset: <https://physionet.org/physiobank/database/eegmmidb/>

⁶ErrP dataset: <https://www.kaggle.com/c/inria-bci-challenge/>

Table 3: DNN model parameters in Fig. 7; $\text{Conv}(h, w)_g^c$ denotes 2D convolution layer with kernel size of (h, w) for output channel of c over group g . $\text{FC}(h)$ denotes fully-connected layer with h output nodes. BN denotes batch normalization.

Classifier \mathcal{C}	Encoder \mathcal{E}	Decoder \mathcal{D}	Nuisance \mathcal{N}	Adversary \mathcal{A}
$\text{FC}(2 Z)$	$\text{Conv}(1, 15)^{50}$	$\text{FC}(20T)$	$\text{FC}(2 Z)$	$\text{FC}(2 Z)$
BN+ReLU	BN+ReLU	ReLU	BN+ReLU	BN+ReLU
$\text{FC}(Y)$	$\text{Conv}(1, 7)^{50}$	$\text{Conv}(C, 1)^{50}$	$\text{FC}(S)$	$\text{FC}(S)$
	BN+ReLU	BN+ReLU		
	$\text{Conv}(1, 3)^{50}$	$\text{Conv}(1, 3)^{50}$		
	BN+ReLU	BN+ReLU		
	$\text{Conv}(C, 1)^{50}_{50}$	$\text{Conv}(1, 7)^{50}$		
	$\text{FC}(Z)$	BN+ReLU		
		$\text{Conv}(1, 15)^{50}$		

- **Faces Basic:** An implanted electrocorticography (ECoG) array dataset for visual stimulus experiments (Miller et al., 2015; 2016).⁷ ECoG arrays were implanted on the subtemporal cortical surface of $|S| = 14$ epilepsy patients. $|Y| = 2$ classes of grayscale images, either faces or houses, were displayed rapidly in random sequence for 400 ms each with black-screen intervals of 400 ms. The ECoG potentials were measured with respect to a scalp reference and ground, at a sampling rate of 1000 Hz. Subjects performed a basic face and house discrimination task. There were 3 sessions for each patient, with 50 house pictures and 50 face pictures in each run, in total 4,100 samples. We use the first $C = 31$ channels to analyze for $T = 400$. Reusing the public dataset requires the ethics statement information.⁸
- **Faces Noisy:** The implanted ECoG arrays dataset for visual stimulus experiments (Miller et al., 2015; 2017). The experiment is similar to Faces Basic dataset, while pictures of faces and houses are randomly scrambled. There are $|S| = 7$ subjects with $C = 39$ channels. Refer ethics statement to reuse the dataset.⁹
- **ASL:** An EMG dataset for finger gesture identification for American Sign Language (ASL) (Günay et al., 2019).¹⁰ $|S| = 5$ healthy, right-handed, subjects participated in experiments with surface EMG (Delsys Inc. Trigno) recorded at 2 kHz from $|C| = 16$ lower-arm muscles. Subjects shaped their right hand into letters and numbers of the ASL posture set presented as pictures on a computer screen ($|Y| = 33$ postures, 3 trials per posture). Dynamic letters ‘J’ and ‘Z’ were omitted, along with the number ‘0’, which is visually the same as the letter ‘O’. The participants were given 2 seconds to form the posture, 6 seconds to maintain it, and 2 seconds to rest between trials. The signal is decimated to be $T = 100$.

A.7 DNN MODEL PARAMETERS

For 2D datasets, we use deep CNN for the encoder \mathcal{E} and decoder \mathcal{D} blocks. For the classifier \mathcal{C} , nuisance estimator \mathcal{N} , and adversary \mathcal{A} , we use a multi-layer perceptron (MLP) having three layers,

⁷Faces dataset: <https://exhibits.stanford.edu/data/catalog/zk881ps0522>

⁸**Ethics statement:** All patients participated in a purely voluntary manner, after providing informed written consent, under experimental protocols approved by the Institutional Review Board of the University of Washington (#12193). All patient data was anonymized according to IRB protocol, in accordance with HIPAA mandate. These data originally appeared in the manuscript “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” published in PLoS Computational Biology in 2016 (Miller et al., 2016).

⁹All patients participated in a purely voluntary manner, after providing informed written consent, under experimental protocols approved by the Institutional Review Board of the University of Washington (#12193). All patient data was anonymized according to IRB protocol, in accordance with HIPAA mandate. These data originally appeared in the manuscript “Face percept formation in human ventral temporal cortex” published in Journal of Neurophysiology in 2017 (Miller et al., 2017).

¹⁰ASL Dataset: <http://hdl.handle.net/2047/D20294523>

whose hidden nodes are doubled from the input dimension. We also use batch normalization (BN) and ReLU activation as listed in Table 3. Note that for a tabular data such as Stress datasets, CNN was replaced with 3-layer MLP having ReLU activation and dropout with a ratio of 20%. Also the MLP classifier was replaced with CNN for 2D input dimension cases such as in the model A. The number of latent dimensions was chosen $|Z| = 64$. When we need to feed S along with 2D data of X into the CNN encoder such as in the model Ds, dimension mismatch poses a problem. We address this issue by using one linear layer to project S into the temporal dimensional space of X and another linear layer to project it into the spatial dimensional space of X . The dot product of those two projected vectors is concatenated as additional channel input. We use $\lambda_* = 0.01$ for the regularization coefficient. We leave hyperparameter exploration to integrate AutoML and AutoBayes as a remaining future work.

A.8 PERFORMANCE RESULTS

The additional results for the all datasets are listed in Table 4. The results suggest that the best inference strategy highly depends on datasets. Specifically, the best model at one dataset does not perform best for different datasets; e.g., the model non-variational Is was best for ASL dataset, while the model variational Ds was best for RSVP dataset. It suggests that we shall consider different inference strategies for each target dataset and AutoBayes provides such an adaptive framework. Also note that reconstruction loss may not be a good indicator to select the graph model. In addition, a huge performance gap between the best and worst models was observed for some datasets. For example, the task accuracy of 76.4% was achieved with model non-variational Dz for Faces (Noisy) dataset, whereas the model variational B offers 51.4%. This implies that we may have a potential risk that one particular model cannot achieve good performance if we do not explore different models.

Table 4: Performance of datasets: the reconstruction loss, the scores of nuisance classification and task classification in variational/non-variational and adversarial/non-adversarial setting.

Dataset	Method	Reconstruction Loss (dB)		Nuisance Classification (%)		Task Classification (%)	
		Non-Variational	Variational	Non-Variational	Variational	Non-Variational	Variational
QMNST	Model A	-51.73	—	—	—	99.02	—
	Model B	-65.68	-61.62	—	—	98.72	99.44
	Model Cs	-66.38	—	13.12	—	99.32	—
	Model Cy	-67.74	—	12.17	—	99.30	—
	Model Ds	-57.14	-41.43	10.55	9.90	99.35	99.23
	Model Dz	-65.04	-66.74	0.44	0.46	99.16	99.27
	Model Es	-65.35	-66.56	11.77	10.51	99.44	99.21
	Model Ez	-65.51	-61.41	2.55	14.95	99.35	99.13
	Model Fs	-57.39	-43.39	14.94	16.50	99.34	99.40
	Model Fz	-65.85	-43.42	1.80	9.03	99.08	99.41
	Model Gs	-64.88	-61.51	9.78	10.25	98.54	98.88
	Model Gz	-65.68	-42.05	9.71	12.36	99.12	98.73
	Model Hs	-66.02	-43.32	15.94	16.56	99.18	99.39
	Model Hz	-65.85	-43.45	13.20	14.70	99.47	99.28
	Model Is	-65.35	-45.41	15.96	18.57	99.46	99.32
	Model Iz	-65.84	-45.46	14.97	15.45	99.54	99.28
	Model Js	-59.02	-57.3	11.41	11.21	99.47	99.39
	Model Jz	-67.96	-61.51	6.44	5.02	98.85	99.46
	Model Ks	-65.51	-63.35	11.59	1.16	99.49	99.10
	Model Kz	-67.33	-61.20	6.32	6.94	99.15	99.15
Stress	Model A	-56.31	—	—	—	85.87	—
	Model B	-66.56	-59.41	—	—	94.79	92.67
	Model Cs	-67.74	—	59.46	—	93.48	—
	Model Cy	-66.56	—	75.77	—	91.93	—
	Model Ds	-61.94	-36.04	59.90	28.37	93.26	83.70
	Model Dz	-66.02	-48.40	81.17	36.21	94.22	79.76
	Model Es	-66.38	-63.35	54.21	79.76	94.00	92.05
	Model Ez	-64.73	-59.25	90.35	91.92	95.02	30.00
	Model Fs	-64.73	-38.68	68.45	40.74	94.07	87.80
	Model Fz	-66.94	-38.57	83.25	5.18	94.92	87.24
	Model Gs	-67.96	-64.73	53.94	25.88	93.61	86.56
	Model Gz	-65.85	-39.16	82.86	69.26	94.11	89.04
	Model Hs	-65.04	-38.47	78.36	72.42	94.72	92.86
	Model Hz	-66.38	-38.37	84.10	71.07	94.57	90.73
	Model Is	-66.74	-47.94	79.51	74.38	94.74	91.94
	Model Iz	-67.96	-47.98	84.46	68.63	94.80	90.52
	Model Js	-67.13	-36.17	79.36	92.47	95.35	30.00
	Model Jz	-66.74	-54.02	86.27	58.59	95.17	86.99
	Model Ks	-68.64	-51.50	73.57	87.33	94.65	86.74
	Model Kz	-66.56	-51.94	85.00	61.84	94.35	86.34
RSVP	Model A	-30.69	—	—	—	93.07	—
	Model B	-34.27	-35.36	—	—	93.06	91.89
	Model Cs	-31.33	—	90.12	—	91.56	—
	Model Cy	-31.57	—	90.38	—	91.54	—
	Model Ds	-35.61	-30.17	91.33	84.77	91.16	93.42
	Model Dz	-35.27	-35.37	92.42	86.84	92.44	92.71
	Model Es	-35.61	-31.44	91.74	90.46	93.23	92.99
	Model Ez	-35.62	-35.52	94.26	93.01	92.65	91.99
	Model Fs	-35.60	-30.17	91.03	90.38	92.15	93.27
	Model Fz	-32.94	-30.16	9.57	9.88	90.21	91.04
	Model Gs	-35.78	-31.24	92.17	92.90	89.83	86.82
	Model Gz	-35.28	-30.34	91.27	90.18	92.15	91.31
	Model Hs	-35.40	-30.18	93.89	91.31	93.05	91.22
	Model Hz	-35.39	-30.18	91.49	89.84	92.65	92.76
	Model Is	-35.37	-30.35	93.37	90.32	92.94	91.60
	Model Iz	-35.37	-30.36	91.36	90.96	91.41	91.92
	Model Js	-36.10	-36.09	92.78	9.92	90.82	92.74
	Model Jz	-35.82	-36.65	93.60	82.62	93.12	92.85
	Model Ks	-35.65	-36.05	90.93	92.86	93.19	90.54
	Model Kz	-35.53	-36.01	91.99	82.10	92.81	93.03

Table 4: Performance of datasets (continued)

Dataset	Method	Reconstruction Loss (dB)		Nuisance Classification (%)		Task Classification (%)	
		Non-Variational	Variational	Non-Variational	Variational	Non-Variational	Variational
MI	Model A	-30.28	—	—	—	55.85	—
	Model B	-32.17	-32.24	—	—	56.32	47.61
	Model Cs	-32.12	—	35.99	—	52.65	—
	Model Cy	-32.15	—	43.60	—	52.98	—
	Model Ds	-31.34	-20.20	74.15	1.14	24.26	24.89
	Model Dz	-32.14	-35.92	4.82	9.01	55.26	51.80
	Model Es	-32.22	-30.90	61.95	0.74	44.74	24.85
	Model Ez	-32.52	-30.82	5.77	8.21	54.12	48.65
	Model Fs	-30.36	-20.35	38.60	0.66	48.90	51.91
	Model Fz	-31.86	-29.77	3.05	0.96	57.83	25.40
	Model Gs	-32.16	-30.07	33.97	0.55	53.01	24.82
	Model Gz	-32.31	-30.06	4.82	0.96	52.61	26.40
	Model Hs	-32.11	-30.08	88.42	57.87	52.68	49.04
	Model Hz	-31.99	-30.02	43.93	1.07	57.21	25.96
	Model Is	-32.27	-30.08	85.55	54.99	55.00	24.26
	Model Iz	-32.35	-30.09	48.49	1.03	53.57	26.03
	Model Js	-30.29	-30.10	49.19	0.80	41.54	24.93
	Model Jz	-32.88	-35.14	43.64	31.10	57.50	44.93
	Model Ks	-30.79	-30.18	81.18	0.77	23.79	25.18
	Model Kz	-32.27	-32.44	29.26	28.31	48.12	48.79
ErrP	Model A	-31.04	—	—	—	69.89	—
	Model B	-41.26	-39.79	—	—	71.81	71.39
	Model Cs	-39.26	—	94.95	—	63.68	—
	Model Cy	-41.51	—	98.98	—	70.07	—
	Model Ds	-39.44	-29.92	98.68	7.69	69.11	69.77
	Model Dz	-42.52	-39.46	97.30	68.93	68.09	75.91
	Model Es	-39.49	-38.91	97.12	92.91	70.01	65.38
	Model Ez	-41.17	-41.98	47.18	99.64	70.91	72.42
	Model Fs	-39.54	-30.00	98.32	6.73	71.45	70.07
	Model Fz	-41.35	-30.10	93.33	8.35	66.71	70.19
	Model Gs	-40.23	-33.96	97.00	0.42	70.85	70.31
	Model Gz	-41.02	-29.94	96.57	98.68	69.23	67.31
	Model Hs	-40.03	-28.32	98.14	98.02	67.85	29.93
	Model Hz	-41.19	-29.90	96.81	97.12	68.81	69.11
	Model Is	-38.09	-30.07	98.26	96.33	59.62	67.31
	Model Iz	-40.54	-29.99	96.21	96.33	70.25	66.95
	Model Js	-40.33	-34.44	98.20	6.07	68.57	68.03
	Model Jz	-42.40	-41.27	99.04	95.13	72.54	69.29
	Model Ks	-38.85	-37.71	98.86	5.77	68.63	69.29
	Model Kz	-42.48	-40.05	98.32	95.01	72.36	69.65

Table 4: Performance of datasets (continued)

Dataset	Method	Reconstruction Loss (dB)		Nuisance Classification (%)		Task Classification (%)	
		Non-Variational	Variational	Non-Variational	Variational	Non-Variational	Variational
Faces Basic	Model A	-29.95	—	—	—	63.30	—
	Model B	-33.68	-30.10	—	—	48.56	51.12
	Model Cs	-32.18	—	80.45	—	64.50	—
	Model Cy	-32.96	—	87.26	—	65.62	—
	Model Ds	-32.99	-30.10	92.23	7.69	62.74	48.08
	Model Dz	-31.68	-23.37	88.70	7.77	66.99	49.28
	Model Es	-31.98	-30.08	92.95	6.73	50.96	53.12
	Model Ez	-31.84	-30.03	38.94	97.60	50.96	51.36
	Model Fs	-33.32	-30.11	96.07	8.09	61.14	62.82
	Model Fz	-32.95	-28.80	49.60	10.02	61.30	61.14
	Model Gs	-32.56	-29.76	91.11	7.05	63.38	49.92
	Model Gz	-33.13	-30.11	85.02	83.41	63.86	64.02
	Model Hs	-32.03	-30.08	98.00	86.22	61.14	64.42
	Model Hz	-33.29	-29.41	91.11	83.81	65.46	61.94
	Model Is	-31.63	-30.11	97.92	94.39	62.34	61.94
	Model Iz	-33.20	-30.06	91.67	89.10	63.94	67.31
	Model Js	-33.28	-30.12	94.87	8.33	51.04	52.23
	Model Jz	-32.21	-29.50	93.83	7.29	65.79	51.28
	Model Ks	-31.12	-29.88	88.94	7.45	51.92	53.85
	Model Kz	-32.69	-30.09	93.43	7.93	51.76	51.84
Faces Noisy	Model A	-30.09	—	—	—	75.94	—
	Model B	-30.35	-30.09	—	—	73.59	51.41
	Model Cs	-30.10	—	95.62	—	75.16	—
	Model Cy	-30.56	—	96.56	—	71.56	—
	Model Ds	-30.22	-27.90	82.34	13.28	74.84	51.72
	Model Dz	-30.11	-30.09	96.09	14.38	76.41	53.91
	Model Es	-30.09	-28.70	91.09	13.28	74.38	52.50
	Model Ez	-30.47	-28.58	21.41	93.75	70.94	52.97
	Model Fs	-30.14	-30.08	95.62	13.75	71.88	75.62
	Model Fz	-29.96	-27.76	27.50	17.03	72.50	72.19
	Model Gs	-28.46	-30.15	93.75	13.91	71.56	52.50
	Model Gz	-30.59	-30.09	94.53	80.94	75.00	75.16
	Model Hs	-30.04	-30.08	98.49	88.59	75.59	69.06
	Model Hz	-30.30	-30.06	95.94	91.09	75.47	76.09
	Model Is	-30.10	-30.04	97.97	96.88	68.91	69.53
	Model Iz	-30.62	-29.86	88.91	87.19	74.06	72.50
	Model Js	-30.08	-28.72	95.69	15.94	65.31	53.59
	Model Jz	-30.57	-30.03	96.62	14.22	71.56	52.66
	Model Ks	-30.29	-30.14	65.62	15.52	54.06	53.44
	Model Kz	-30.12	-28.45	94.84	12.66	76.56	54.23
ASL	Model A	-24.22	—	—	—	41.69	—
	Model B	-23.89	-24.08	—	—	3.03	37.80
	Model Cs	-24.07	—	93.63	—	38.35	—
	Model Cy	-24.14	—	94.63	—	38.28	—
	Model Ds	-24.07	-24.08	93.74	94.29	39.23	41.32
	Model Dz	-24.47	-24.69	95.99	95.10	43.83	40.89
	Model Es	-24.07	-24.07	94.00	93.60	40.07	40.38
	Model Ez	-24.96	-24.10	43.16	85.45	43.56	37.23
	Model Fs	-24.07	-24.08	93.93	97.58	38.75	42.27
	Model Fz	-24.08	-24.08	9.99	10.79	28.25	42.16
	Model Gs	-24.07	-24.08	94.45	93.81	38.81	39.83
	Model Gz	-24.50	-24.81	95.69	94.76	47.43	43.32
	Model Hs	-25.10	-24.08	96.61	94.26	49.30	36.39
	Model Hz	-24.87	-24.08	94.77	94.20	48.31	37.33
	Model Is	-24.87	-24.08	96.54	94.37	51.12	38.31
	Model Iz	-24.74	-25.03	95.81	93.98	49.47	38.45
	Model Js	-24.07	-24.11	93.64	97.09	38.39	36.77
	Model Jz	-24.09	-24.11	14.27	96.44	6.24	37.25
	Model Ks	-24.11	-24.05	93.10	16.26	38.07	8.19
	Model Kz	-24.22	-24.22	12.34	95.83	3.03	37.75