Bayesian EDDI: Sequential Variable Selection with Bayesian Partial VAE

Chao Ma [1 *] **Wenbo Gong** [1 *] **Sebastian Tschiatschek** [2] **Sebastian Tschiatschek** [1 2] **Sebastian Nowozin** [2 3]
José Miguel Hernández-Lobato [1 2] **Cheng Zhang** [2]

[*]Equal contribution

Abstract

Obtaining more relevant information enables better decision making, but may be costly. Optimal sequential decision making allows us to trade off the desire to make good decisions by acquiring further information with the cost of performing that acquisition. To this end, we propose a principled framework, named *EDDI* (Efficient Dynamic Discovery of high-value Information), based on the theory of Bayesian experimental design. In EDDI we propose a novel *partial variational autoencoder* (Partial VAE), to efficiently handle missing data with different missing patterns. Additionally, we extend the VAE based framework to Bayesian treatment of the weights, which obtains better performance in small data regime. EDDI then combines it with an acquisition function that maximizes expected information gain on a set of target variables at each step.

1 Introduction

Imagine that a person walks into a hospital with a broken arm. The first question from health-care personnel would be: "How did you break the arm?" instead of "Do you have a cold?", because the answer reveals relevant information for this patient. Human experts dynamically acquire information based on the current understanding of the situation in a sequential manner . Automating this human expertise of asking relevant questions is difficult. In other applications such as online questionnaires, for example, most existing online questionnaire system either present exhaustive questions (Lewenberg et al., 2017; Shim et al., 2018) or use extremely time-consuming human labeling work to manually build a decision tree for a reduced number of questions (Zakim et al., 2008). This wastes the valuable time of experts or users (patients). An automated solution for personalized dynamic acquisition of information has great potential to save much of this time in many real-life applications.

What are the technical challenges to building a sequential intelligent information acquisition system? *Missing data is a key issue*: taking the questionnaire scenario as an example, there are two types of missing data problems. Firstly, for each subject, at any point in time we only observe a small subset of answers yet have to reason about possible answers for the remaining questions. We thus need an accurate probabilistic model that can perform inference given a variable subset of observed answers. Secondly, in many real-life scenarios, the number of initially available subjects is often limited. Can the system perform robustly under small data? *Another key problem is deciding what to ask next*: this requires assessing the value of each possible question or measurement, the exact computation of which is intractable. However, compared to current active learning methods we select individual features, not instances; therefore, existing methods are not applicable. In addition, these traditional methods are often not scalable to the large volume of data available in many practical cases (Settles, 2012).

We propose the EDDI (Efficient Dynamic Discovery of high-value Information) framework as a scalable sequential information acquisition system. We assume that information acquisition is always associated with cost. Given a task, we dynamically decide which piece of information to acquire next. We contribute:

- A new partial amortized inference method for generative modeling under partial observations.We name it *Partial VAE* (Partial Variational AutoEncoder). (Section 3.1).
- A novel fully-Bayesian treatment of Partial VAE to incorporate small datasets, where variational inference and MCMC are combined for inference (Section 3.1).
- An information theoretic acquisition function with an efficient approximation, yielding a novel variable-wise

[1]Department of Engineering, University of Cambridge, Cambridge, UK [2]Microsoft Research Cambridge, Cambridge, UK [3]now at Google AI, Berlin, Germany (contributed to this paper during his stay at Microsoft Research Cambridge). Correspondence to: Cheng Zhang <Cheng.Zhang@microsoft.com>.

Real-world Sequential Decision Making workshop at ICML 2019.

active learning method (Section 3.2).

2 Related work

Our proposed framework - EDDI - requires the missing data model to handle partial observations and sequential decision making objective to acquire information actively. We thus review highly related work on these two topics.

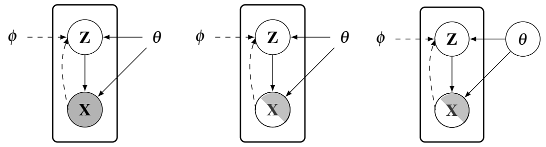
Missing data imputation. Prediction based methods have shown advantages for missing value imputation (Rubin, 1976; Dempster et al., 1977; Scheffer, 2002). Efficient matrix factorization (MF) based methods are often applied (Keshavan et al., 2010; Jain et al., 2010; Salakhutdinov & Mnih, 2008). The probabilistic MF has been used in the active variable selection framework (Lewenberg et al., 2017). However, they do not scale to large volumes of data and thus are usually not applicable in real-world applications. Nazabal et al. (2018) use zero imputation (ZI) for amortized inference for both training and test sets with missing data entries. The drawback of ZI is that it introduces bias when the data are not missing completely at random. It also produces poor uncertainty estimates since it does not distinguish between observed data and imputed data.

Active Learning. Traditional active learning aims to select the *next data point* to label. Information theoretical approaches have shown promising results with different acquisition functions (MacKay, 1992; McCallumzy & Nigamy, 1998; Houlsby et al., 2011). Little work has dealt with missing values within instances. Zheng & Padmanabhan (2002) deal with missing data values by imputing with traditional non-probabilistic methods (Little & Rubin, 1987) first. We note that these methods assume that the data are fully observed, and the acquisition decision is instance wise. Different from the approaches, our proposed framework performs *variable-wise active learning* for *each* instance.

3 Method

Problem formulation In this paper we address the following sequential active variable selection problem. Let $\mathbf{x} = [x_1, \dots, x_{|I|}]$ be a set of random variables with probability density $p(\mathbf{x})$. At each step, let a subset of the variables $\mathbf{x}_O$, $O \subset I$, be observed while the variables $\mathbf{x}_U$, $U = I \setminus O$, are unobserved. We assume that we can query the value of variables x_i for $i \in U$. The goal of sequential active variable selection at each step is to query a sequence of variables in U in order to predict a quantity of interest $f(\mathbf{x})$, as accurately as possible, where $f(\cdot)$ can be any (random) function. This problem, in the simplified myopic setting, can be formalized as that of proposing the next variable x_{i^*} to be queried by maximizing a reward function R, i.e.

$$i^* = \arg\max_{i \in U} R(i \mid \mathbf{x}_O), \tag{1}$$



(a) VAE (b) Partial VAE (c) Bayesian PVAE

Figure 1: Graphical representation

where $R(i \mid \mathbf{x}_O)$ quantifies the merit of our prediction of $f(\cdot)$ given $\mathbf{x}_0$ and x_i. Furthermore, the reward can quantify other properties, e.g. the cost of acquiring x_i.

3.1 Partial Amortization of Inference Queries

As discussed before, to enable sequential active feature selection, we need a scalable model to handle missing values. We present Partial VAE (Figure 1(b)) which enables amortized inference for partially observed data and present a full Bayesian extension, Bayesian Partial VAE (Figure 1(c)).

From VAE to Partial VAE. A VAE (Figure 1(a)) defines a generative model in which the data $\mathbf{x}$ is generated from latent variables $\mathbf{z}$, defined as $p(\mathbf{x}, \mathbf{z}; \theta) = \prod_n p_\theta(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)$, where $p_\theta(\mathbf{x}_n|\mathbf{z}_n)$, is realized by a deep neural network. To approximate the posterior $p_\theta(\mathbf{z}_n|\mathbf{x}_n)$, VAEs use *amortized* variational inference. Specifically, it uses an encoder, which is another neural network with the data $\mathbf{x}_n$ as input to produce a variational approximation of the posterior $q(\mathbf{z}_n|\mathbf{x}_n; \phi)$. As traditional variational inference, VAE is trained by maximizing an evidence lower bound (ELBO).

VAEs are not directly applicable when data points contain missing values. Non-armortized variational inference could be applied, but will be too impractical due to the space of possible missing patterns is huge. However, note that in a VAE, $p(\mathbf{x}|\mathbf{z})$ is factorized, i.e.

$$p(\mathbf{x}|\mathbf{z}) = \prod_i p_i(\mathbf{x}_i|\mathbf{z}). \tag{2}$$

This implies that given $\mathbf{z}$, the observed variables $\mathbf{x}_O$ are conditionally independent of $\mathbf{x}_U$. Therefore,

$$p(\mathbf{x}_U|\mathbf{x}_O, \mathbf{z}) = p(\mathbf{x}_U|\mathbf{z}), \tag{3}$$

and inferences about $\mathbf{x}_U$ can be reduced to inference about $\mathbf{z}$. Hence, the key object of interest in this setting is $p(\mathbf{z}|\mathbf{x}_O)$, i.e., the posterior over the shared latent variables $\mathbf{z}$ given the observed variables $\mathbf{x}_O$. Once we obtain $\mathbf{z}$, computing $\mathbf{x}_U$ is straightforward. To approximate $p(\mathbf{z}|\mathbf{x}_O)$, we introduce an auxiliary variational inference network $q(\mathbf{z}|\mathbf{x}_O)$ and define a partial variational lower bound $\mathscr{L}_{partial}$,

$$\begin{aligned} \log p(\mathbf{x}_O) &\geq \log p(\mathbf{x}_O) - D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x}_O) \| p(\mathbf{z}|\mathbf{x}_O)) \\ &= \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}_O)}[\log p(\mathbf{x}_O|\mathbf{z}) + \log p(\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x}_O)] \\ &\equiv \mathscr{L}_{partial}. \end{aligned} \tag{4}$$

We call the the inference net for $q(\mathbf{z}|\mathbf{x}_O)$ the *partial inference net* since it takes a set of partially observed variables $\mathbf{x}_O$ whose dimensionality may vary among different data points. Specifying $q(\mathbf{z}|\mathbf{x}_O)$ requires distributions for any partition $\{O, U\}$ of I.

From Partial VAE to Bayesian Partial VAE Both Partial VAE and VAE utilize deep neural networks for the generative model. It is flexible but requires a large volume of data for training. In many real-world applications, only small amounts of training data are available, resulting in that the model overfits the training data. To address this, we propose a fully Bayesian scheme for the generative model, naming Bayesian Partial VAE. In particular, we assume the decoder weights, $\mathbf{w}$, are also random variables with the prior distribution $p(\mathbf{w})$. Approximate inference for both $\mathbf{z}$ and $\mathbf{w}$ is required. Mean-field amortized inference is scalable but less accurate comparing to MCMC due to the mean-field assumption. In this case, z is a local variable which scales with the number of data N, whereas w is a low dimensional global variable. We thus propose to combine variational inference for $\mathbf{z}$ and MCMC inference for $\mathbf{w}$ by performing the following two steps recursively:

- Variational inference stage: Conditioned on the previous MCMC sample of $\mathbf{w}$, we optimize the $\mathscr{L}_{partial}$ w.r.t the encoder $q(\mathbf{z}|\mathbf{x}_O)$, with $\mathbf{w}$ fixed.
- MCMC stage: We use Stochastic Gradient Hamiltonian Monte Carlo (SG-HMC) (Chen et al., 2014) to sample $\mathbf{w}$. This requires computing the gradient $\nabla_{\mathbf{w}} \log p(\mathbf{x}_O, \mathrm{w})$, which is approximated by $\mathscr{L}_{partial} + \log p(\mathbf{w})$ from the variational inference stage.

Amortized Inference with partial observations. For both Partial VAE and Bayesian Partial VAE , inference under partial observations requires the inference net $q(\mathbf{z}|\mathbf{x}_O)$ to be capable to handle arbitrary set of observed data, and sharing parameters across these different sized sets of observations for amortization.

Inspired by the *Point Net (PN)* approach for point cloud classification (Qi et al., 2017; Zaheer et al., 2017), we specify the approximate distribution $q(\mathbf{z}|\mathbf{x}_O)$ by a *permutation invariant set function encoding*, given by:

$$\mathbf{c}(\mathbf{x}_O) := g(h(\mathbf{s}_1), h(\mathbf{s}_2), ..., h(\mathbf{s}_{|O|})), \tag{5}$$

where $\mathbf{s}_d$ carries the information of the input of d-th observed variable, and $|O|$ is the number of observed variables. In particular, $\mathbf{s}_d$ contains the information about the identity of the input $\mathbf{e}_d$ and the corresponding input value x_d. There are many ways to define $\mathbf{e}_d$, such as the coordinates of observed pixels for images. In this work, we treat $\mathbf{e}$ as an unknown embedding, which is optimized during training.

There are also different ways to construct $\mathbf{s}_d$. A common choice is concatenation, $\mathbf{s}_d = [\mathbf{e}_d, x_d]$, which is commonly used in computer vision applications (Qi et al., 2017). We refer to this setting using concatenation as the *Pointnet (PN)* specification of Partial VAE. However, the construction of $\mathbf{s}_d$ can be more flexible.

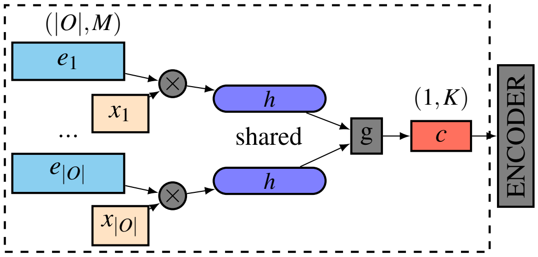


Figure 2: Illustration of Partial VAE encoder architecture (PNP setting)

We propose to construct $\mathbf{s}_d = \mathbf{e}_d * x_d$ using element-wise multiplication, shown in Figure 2. We show that this formulation generalizes naive Zero Imputation (ZI) VAE (Nazabal et al., 2018) (cf. Appendix B.1). We refer to the multiplication setting as *Pointnet Plus (PNP)* specification of Partial VAE. The theoretical consideration of relating ZI to PNP is presented in Appendix B.1.

We can then use a neural network $h(\cdot)$ to map input $\mathbf{s}_d$ to $\mathbb{R}^K$, where and K is the latent space size. The key to the PNP structure is the permutation invariant aggregation operation $g(\cdot)$, such as max-pooling or summation. In this way, the mapping $\mathbf{c}(\mathbf{x}_O)$ is invariant to permutations of elements of $\mathbf{x}_O$, and $\mathbf{x}_O$ can have arbitrary length. Finally, the fixed-size code $\mathbf{c}(\mathbf{x}_O)$ is fed into a ordinary neural network, that transforms the code into the statistics of a multivariate Gaussian distribution to approximate $p(\mathbf{z}|\mathbf{x}_O)$.

3.2 Efficient Dynamic Discovery of High-value Information

We now cast the sequential active variable selection problem (1) as adaptive Bayesian experimental design at each step, utilizing $p(\mathbf{x}_U|\mathbf{x}_O)$ inferred by the Partial VAE. Algorithm 1 summarize the EDDI framework.

Information Reward. We designed a variable selection acquisition function in an information theoretic way following Bayesian experimental design (Lindley, 1956; Bernardo, 1979). For a given task, we may be interested in statistics of some variables $\mathbf{x}_\phi$, where $\mathbf{x}_\phi \subset \mathbf{x}_U$. Given a new instance (user), assume we have observed $\mathbf{x}_O$ so far for this instance, and we need to select the next variable x_i (an element of $\mathbf{x}_{U \setminus \phi}$) to observe. Following Bernardo (1979), We select x_i by maximizing:

$$R(i, \mathbf{x}_O) = \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_O)} D_{\mathrm{KL}} \left[p(\mathbf{x}_\phi|\mathbf{x}_i, \mathbf{x}_O) \, \| \, p(\mathbf{x}_\phi|\mathbf{x}_O) \right]. \tag{6}$$

In our paper, we mainly consider the case that a subset of interesting observations represents the statistics of interest $\mathbf{x}_\phi$. Sampling $\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)$ is approximated by $\mathbf{x}_i \sim \hat{p}(\mathbf{x}_i|\mathbf{x}_o)$,

Algorithm 1 EDDI: Algorithm Overview

```
Require: Training dataset $\mathbf{X}$, which is partially observed;
    Test dataset $\mathbf{X}^*$ with no observations collected yet; Indices $\phi$ of target variables.
Train (Bayesian) Partial VAE by optimizing partial variational bound(cf. Section 3.1)
Actively acquire feature value $x_i$ to estimate $\mathbf{x}_\phi^*$ for each test point (cf. Section 3.2)
  for each test instance do
      $\mathbf{x}_O \leftarrow \emptyset$ (no variable value has been observed for any test point)
      repeat
          Choose variable $x_i$ from $U \setminus \phi$ to maximize the information reward (Equation (9))
          $\mathbf{x}_O \leftarrow x_i \cup \mathbf{x}_O$
      until Stopping criterion reached (e.g. the time budget)
  end for
```

where $\hat{p}(\mathbf{x}_i|\mathbf{x}_o)$ can be obtained by (Bayesian) Partial VAE. It is implemented by sampling $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}_o)$, then $\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{z})$. The same applies for $p(\mathbf{x}_i, \mathbf{x}_\phi|\mathbf{x}_o)$ in Equation (8).

Efficient approximation of the Information reward. The Partial VAE allows us to sample $\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)$. However, the KL term in Equation (6),

$$
\begin{aligned}
&D_{KL}\left[p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{x}_\phi|\mathbf{x}_o)\right] \qquad (7)\\
&= -\int_{\mathbf{x}_\phi} p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)\log\frac{p(\mathbf{x}_\phi|\mathbf{x}_o)}{p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)},
\end{aligned}
$$

is intractable since both $p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)$ and $p(\mathbf{x}_\phi|\mathbf{x}_o)$ are intractable. For high dimensional $\mathbf{x}_\phi$, entropy estimation could be difficult. The entropy term $\int_{\mathbf{x}_\phi} p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)\log p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)$ depends on i hence cannot be ignored. In the following, we show how to approximate this expression.

Our proposal is based on the observation that analytic solutions of KL-divergences are available under specific variational distribution families of $q(\mathbf{z}|\mathbf{x}_O)$ (such as the Gaussian distribution commonly used in VAEs). Instead of calculating information reward in $\mathbf{x}$ space, we have shown that one can equivalently perform calculations in $\mathbf{z}$ space (cf. Appendix A.1):

$$
\begin{aligned}
R(i,\mathbf{x}_o) =& \mathbb{E}_{\mathbf{x}_i\sim p(\mathbf{x}_i|\mathbf{x}_o)} D_{KL}\left[p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_o)\right] - \qquad (8)\\
&\mathbb{E}_{\mathbf{x}_\phi,\mathbf{x}_i\sim p(\mathbf{x}_\phi,\mathbf{x}_i|\mathbf{x}_o)} D_{KL}\left[p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_o)\right].
\end{aligned}
$$

Note that Equation (8) is exact. Additionally, we use the partial VAE approximation $p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o) \approx q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o)$, $p(\mathbf{z}|\mathbf{x}_o) \approx q(\mathbf{z}_i|\mathbf{x}_o)$ and $p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o) \approx q(\mathbf{z}_i|\mathbf{x}_i,\mathbf{x}_o)$. This leads to the final approximation of the information reward:

$$
\begin{aligned}
\hat{R}(i,\mathbf{x}_o) =& \mathbb{E}_{\mathbf{x}_i\sim \hat{p}(\mathbf{x}_i|\mathbf{x}_o)} D_{KL}\left[q(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_o)\right] - \qquad (9)\\
&\mathbb{E}_{\mathbf{x}_\phi,\mathbf{x}_i\sim \hat{p}(\mathbf{x}_\phi,\mathbf{x}_i|\mathbf{x}_o)} D_{KL}\left[q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_o)\right].
\end{aligned}
$$

With this approximation, the divergence between $q(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)$ and $q(\mathbf{z}|\mathbf{x}_o)$ can often computed analytically in the (Bayesian) Partial VAE setting, for example, under Gaussian parameterization. The only Monte Carlo sampling required is the one set of samples $\mathbf{x}_\phi, \mathbf{x}_i \sim p(\mathbf{x}_\phi, \mathbf{x}_i|\mathbf{x}_o)$ that can be shared across different KL terms in Equation (9).

4 Experiments

In this section, we evaluate our EDDI framework on both UCI benchmark and real-life intensive care data. We use EDDI to refer to our framework with Partial VAE and use Bayesian EDDI (BEDDI) when Bayesian Patial VAE is used in this section. Both EDDI and BEDDI demonstrate significant superior performance on prediction accuracy at all steps in the sequential information acquisition.

4.1 Active variable selection with small UCI set

In this section, we demonstrate the performance of EDDI and BEDDI with Boston Housing dataset from UCI benchmark datasets (Dheeru & Karra Taniskidou, 2017). In this experiment, we treat different variables of the same instances as different data points that can be acquired separately. To compare the performance under different amount of training data, We train Partial VAE and Bayesian Partial VAE with random subset of training data with different sizes. After training under each subset of the training data , the EDDI and Bayesian EDDI are used to perform active variable selection. To compare the different strategies, we use the area under the information curve (AUIC), $\sum_t -\log p(\mathbf{x}_\phi|\mathbf{x}_{O_t})$, where the information curve $p(\mathbf{x}_\phi|\mathbf{x}_{O_t})$ is defined as predictive likelihood of target variables after t acquisitions. Smaller AUIC value (could be or negative) indicates better performance. The AUIC performances of EDDI and Bayesian EDDI are compared as the number of observations are gradually increased.

Results are shown in Figure 3. In Figure 3 (a), we plot curves of active learning AUIC on Boston Housing dataset against the size of data set used in the training phase. Both EDDI and Bayesian EDDI consistently outperforms random active feature selection strategy (denoted as RAND). More importantly, Fig 3 shows that Bayesian EDDI is more robust and always outperforms EDDI across different sizes of the training dataset. Specifically, we further visualize the active variable selection process under two sizes of training set. We track the predictive negative likelihoods during active acquisitions. When the training set is relatively small (Fig 3 (b)), EDDI suffers from severe overfitting, while Bayesian EDDI performs better. When the training set is sufficient (Figure 3 (c)), Bayesian EDDI and EDDI obtains better performance. In Figure (c), we can see that using EDDI,a personalized sequential feature selection framework, we can obtain the same prediction performance using only about 30% of the data.

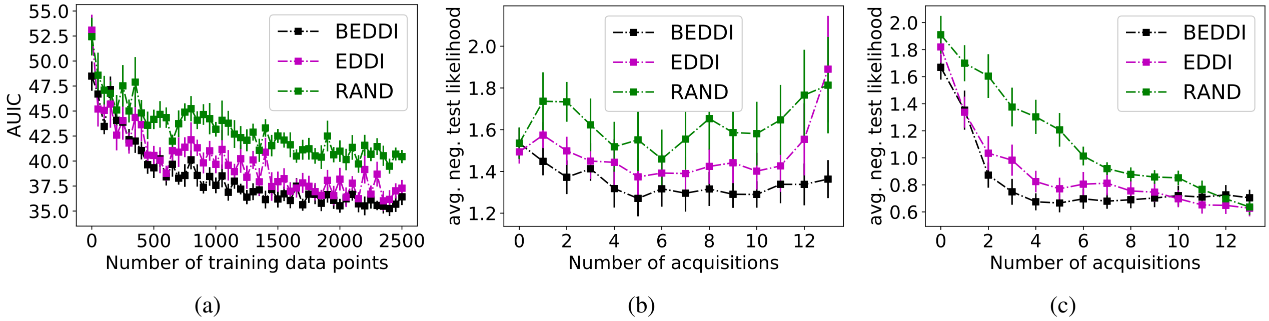


Figure 3: **(a)**: Curves of active learning AUIC on Boston Housing dataset against the size of training set. BEDDI: EDDI with Bayesian partial VAE; EDDI: EDDI with partial VAE; RAND: random selection strategy (with parital VAE). **(b)-(c)** Information curve with 50, and 2500 training data values, respectively. The x-axis shows the number of features that has been acquired and the y-axis shows the negative log likelihood for the prediction.

4.2 Risk assessment with MIMIC-III

We apply EDDI to risk assessment tasks using the Medical Information Mart for Intensive Care (MIMIC III) database ((Johnson et al., 2016)). MIMIC III is the most extensive publicly available clinical database, containing real-world records from over 40,000 critical care patients with 60,000 ICU stays. The risk assessment task is to predict the final mortality. We preprocess the data for this task following Harutyunyan et al. (2017) [1]. This results in a dataset of 21139 patients. We treat the final mortality of a patient as a Bernoulli variable. For our task, we focus on variable selection, which corresponds to medical instrument selection. We thus further process the time series variables into static variables based on temporal averaging.

Since MIMIC-III contains sufficient data, we only focus on EDDI. We compare the performance of EDDI, using four different Partial VAE settings (our PNP-partial VAE, the original PointNet approach, Zero Imputing, and Zero Imputing with masks as input), with three baselines. The first baseline is the *random feature selection strategy (denoted as RAND)* which randomly picks the next variable to observe. The second baseline is the *single best strategy (denoted as SING)* which finds a single fixed global optimal order of selecting variables. This order is then applied to all data points. SING uses the objective function as in Equation (9) to find the optimal ordering by averaging over all the data.

Figure 4 shows the information curve of different strategies, using PNP based Partial VAE as an example. Table 1 shows the average ranking of AUIC with different settings. In this application, EDDI significantly outperforms other variable selection strategies in all different settings of Partial VAE, and PNP based setting performs best.

[1] https://github.com/yerevann/mimic3-benchmarks

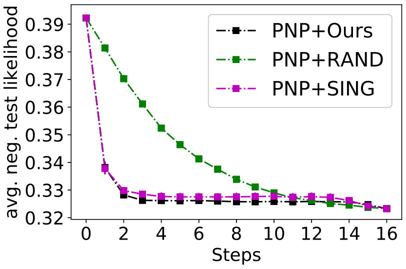


Figure 4: Information curves of active variable selection on risk assessment task on MIMIC III with PNP setting.

Method	EDDI	Random	Single best
ZI	8.83 (0.01)	7.97 (0.02)	9.83 (0.01)
ZI-m	4.91 (0.01)	7.00 (0.01)	5.91 (0.01)
PN	4.96 (0.01)	6.62 (0.01)	5.96 (0.01)
PNP	**4.39 (0.01)**	6.18 (0.01)	5.39 (0.01)

Table 1: Average ranking on AUIC of MIMIC III

5 Conclusion

In this paper, we present EDDI, a novel and efficient framework for dynamic active variable selection for each instance. Within the EDDI framework, we propose Partial VAE which performs amortized inference to handle missing data. We also extend it to Bayesian Partial VAE which can be used even when a small amount of training data is available. Based on (Bayesian) Partial VAE, we design a variable wise acquisition function for EDDI and derive corresponding approximation method. EDDI has demonstrated its effectiveness on active variable selection tasks across multiple real-world applications. In the future, we would extend the EDDI framework to handle more complicated scenarios, such as time-series, or the cold-start situation.

References

Bernardo, J. M. Expected information as expected utility. *The Annals of Statistics*, pp. 686–690, 1979.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691, 2014.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pp. 1–38, 1977.

Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Harutyunyan, H., Khachatrian, H., Kale, D. C., and Galstyan, A. Multitask learning and benchmarking with clinical time series data. *arXiv preprint arXiv:1703.07771*, 2017.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Jain, P., Meka, R., and Dhillon, I. S. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, 2010.

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 2010.

Lewenberg, Y., Bachrach, Y., Paquet, U., and Rosenschein, J. S. Knowing what to ask: A bayesian active learning approach to the surveying problem. In *AAAI*, pp. 1396–1402, 2017.

Lindley, D. V. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pp. 986–1005, 1956.

Little, R. and Rubin, D. Statistical analysis with missing data. Technical report, J. Wiley, 1987.

MacKay, D. J. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

McCallumzy, A. K. and Nigamy, K. Employing em and pool-based active learning for text classification. In *International Conference on Machine Learning*, pp. 359–367. Citeseer, 1998.

Nazabal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. Handling incomplete heterogeneous data using vaes. *arXiv preprint arXiv:1807.03653*, 2018.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.

Ranganath, R., Tran, D., and Blei, D. Hierarchical variational models. In *International Conference on Machine Learning*, pp. 324–333, 2016.

Rubin, D. B. Inference and missing data. *Biometrika*, 63(3): 581–592, 1976.

Salakhutdinov, R. and Mnih, A. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *International conference on Machine learning*, pp. 880–887. ACM, 2008.

Scheffer, J. Dealing with missing data. 2002.

Settles, B. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

Shim, H., Hwang, S. J., and Yang, E. Joint active feature acquisition and classification with variable-size set encoding. In *Advances in Neural Information Processing Systems*, 2018.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *Advances in Neural Information Processing Systems*, pp. 3394–3404, 2017.

Zakim, D., Braun, N., Fritz, P., and Alscher, M. D. Underutilization of information and knowledge in everyday medical practice: Evaluation of a computer-based solution. *BMC Medical Informatics and Decision Making*, 8 (1):50, 2008.

Zheng, Z. and Padmanabhan, B. On active learning for data acquisition. In *International Conference on Data Mining*, pp. 562–569. IEEE, 2002.

A Additional Derivations

A.1 Information reward approximation

In our paper, given the VAE model $p(\mathbf{x}|z)$ and a partial inference network $q(\mathbf{z}|\mathbf{x}_o)$, the experimental design problem is formulated as maximization of the information reward:

$$R(i,\mathbf{x}_o) = \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)}[D_{KL}(p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{x}_\phi|\mathbf{x}_o))]$$

Where $p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o) = \int_{\mathbf{z}} p(\mathbf{x}_\phi|\mathbf{z})q(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)$, $p(\mathbf{x}_\phi|\mathbf{x}_o) = \int_{\mathbf{z}} p(\mathbf{x}_\phi|\mathbf{z})q(\mathbf{z}|\mathbf{x}_o)$ and $q(\mathbf{z}|\mathbf{x}_o)$ are approximate condition distributions given by partial VAE models. Now we consider the problem of directly approximating $R(i,\mathbf{x}_o)$.

Applying the chain rule of KL-divergence, we have:

$$\begin{aligned}
&D_{KL}(p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{x}_\phi|\mathbf{x}_o))\\
&= D_{KL}(p(\mathbf{x}_\phi,\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{x}_\phi,\mathbf{z}|\mathbf{x}_o))\\
&- \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)}\left[D_{KL}(p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_o))\right],
\end{aligned}$$

Using again the KL-divergence chain rule on $D_{KL}(p(\mathbf{x}_\phi,\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{x}_\phi,\mathbf{z}|\mathbf{x}_o))$, we have:

$$\begin{aligned}
&D_{KL}(p(\mathbf{x}_\phi,\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{x}_\phi,\mathbf{z}|\mathbf{x}_o))\\
&= D_{KL}(p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_o)) + D_{KL}(p(\mathbf{x}_\phi|\mathbf{z},\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{x}_\phi|\mathbf{z},\mathbf{x}_o))\\
&= D_{KL}(p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_o)) + D_{KL}(p(\mathbf{x}_\phi|\mathbf{z})||p(\mathbf{x}_\phi|\mathbf{z}))\\
&= D_{KL}(p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_o)).
\end{aligned}$$

The KL-divergence term in the reward formula is now rewritten as follows,

$$\begin{aligned}
&D_{KL}(p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{x}_\phi|\mathbf{x}_o))\\
&= D_{KL}(p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_o))\\
&- \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)}\left[D_{KL}(p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_o))\right].
\end{aligned}$$

One can then plug in the partial VAE inference approximation:

$$\begin{aligned}
&p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o) \approx q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o),\\
&p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o) \approx q(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o), \quad p(\mathbf{z}|\mathbf{x}_o) \approx q(\mathbf{z}|\mathbf{x}_o)
\end{aligned}$$

Finally, the information reward is now approximated as:

$$\begin{aligned}
&R(i,\mathbf{x}_o)\\
&\approx \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)}[D_{KL}(q(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_o))]\\
&- \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)}\mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)}\left[D_{KL}(q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_o))\right]\\
&= \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)}[D_{KL}(q(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_o))]\\
&- \mathbb{E}_{\mathbf{x}_\phi,\mathbf{x}_i \sim p(\mathbf{x}_\phi,\mathbf{x}_i|\mathbf{x}_o)}\left[D_{KL}(q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_o))\right] = \hat{R}(i,\mathbf{x}_o).
\end{aligned}$$

This new objective tries to maximize the shift of belief on latent variables $\mathbf{z}$ by introducing $\mathbf{x}_i$, while penalizing the information that cannot be absorbed by $\mathbf{x}_\phi$ (by the penalty term $D_{KL}(q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o)||q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_o))$). Moreover, it is more computationally efficient since one set of samples $\mathbf{x}_\phi,\mathbf{x}_i \sim p(\mathbf{x}_\phi,\mathbf{x}_i|\mathbf{x}_o)$ can be shared across different terms, and the KL-divergence between common parameterizations of encoder (such as Gaussians and normalizing flows) can be computed exactly without the need for approximate integrals. Note also that under approximation

$$p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o) \approx q(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o), \quad p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o) \approx q(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o), \quad p(\mathbf{z}|\mathbf{x}_o) \approx q(\mathbf{z}|\mathbf{x}_o)$$

, sampling $\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)$ is approximated by $\mathbf{x}_i \sim \hat{p}(\mathbf{x}_i|\mathbf{x}_o)$, where $\hat{p}(\mathbf{x}_i|\mathbf{x}_o)$ is defined by the following process in Partial VAE. It is implemented by first sampling $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}_o)$, and then $\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{z})$. The same applies for $p(\mathbf{x}_i,\mathbf{x}_\phi|\mathbf{z})$.

B Additional Theoretical Contributions

B.1 Zero imputing as a Point Net

Here we present how the zero imputing (ZI) and PointNet (PN) approaches relate.

Zero imputation with inference net In ZI, the natural parameter of λ (e.g., Gaussian parameters in variational autoencoders) is approximated using the following neural network:

$$f(\mathbf{x}) := \sum_{l=1}^{L} w_l^{(1)} \sigma(\mathbf{w}_l^{(0)} \mathbf{x}^T)$$

,

where L is the number of hidden units, $\mathbf{x}$ is the input image with x_i be the value of the i^{th} pixel. To deal with partially observed data $\mathbf{x} = \mathbf{x}_o \cup \mathbf{x}_u$, ZI simply sets all $\mathbf{x}_u$ to zero, and use the full inference model $f(\mathbf{x})$ to perform approximate inference.

PointNet parameterization The PN approach approximates the natural parameter λ by a permutation invariant set function

$$g(h(\mathbf{s}_1), h(\mathbf{s}_2), ..., h(\mathbf{s}_O)),$$

where $\mathbf{s}_i = [x_i, \mathbf{e}_i]$, $\mathbf{e}_i$ is the I dimensional embedding/ID/location vector of the i^{th} pixel, $g(\cdot)$ is a symmetric operation such as max-pooling and summation, and $h(\cdot)$ is a nonlinear feature mapping from $\mathbb{R}^{I+1}$ to $\mathbb{R}^K$ (we will always refer h as *feature maps*). In the current version of the partial-VAE implementation, where Gaussian approximation is used, we set $K = 2H$ with H being the dimension of latent variables. We set g to be the element-wise summation operator, i.e. a mapping from $\mathbb{R}^{KO}$ to $\mathbb{R}^K$ defined by:

$$g(h(\mathbf{s}_1), h(\mathbf{s}_2), ..., h(\mathbf{s}_O)) = \sum_{i \in O} h(\mathbf{s}_i).$$

This parameterization corresponds to products of multiple Exp-Fam factors $\prod_{i \in O} \exp\{-\langle h(\mathbf{s}_i), \Phi \rangle\}$.

From PN to ZI To derive the PN correspondence of the above ZI network we define the following PN functions:

$$h(\mathbf{s}_i) := \mathbf{e}_i * x_i$$

$$g(h(\mathbf{s}_1), h(\mathbf{s}_2), ..., h(\mathbf{s}_O)) := \sum_{k=1}^{I} \theta_k \sigma(\sum_{i \in O} h_k(\mathbf{s}_i)),$$

where $h_k(\cdot)$ is the k^{th} output feature of $h(\cdot)$. The above PN parameterization is also permutation invariant; setting $L = I$, $\theta_l = w_l^{(1)}, (\mathbf{w}_l^{(0)})_i = (\mathbf{e}_i)_l$ the resulting PN model is equivalent to the ZI neural network.

Generalizing ZI from PN perspective In the ZI approach, the missing values are replaced with zeros. However, this ad-hoc approach does not distinguish missing values from actual observed zero values. In practice, being able to distinguish between these two is crucial for improving uncertainty estimation during partial inference. One the other hand, we have found that PN-based partial VAE experiences difficulties in training. To alleviate both issues, we proposed a generalization of the ZI approach that follows a PN perspective. One of the advantages of PN is setting the *feature maps* of the unobserved variables to zero instead of the related weights. As discussed before, these two approaches are equivalent to each other only if the factors are linear. More generally, we can parameterize the PN by:

$$h^{(1)}(\mathbf{s}_i) := \mathbf{e}_i * x_i$$

$$h^{(2)}(h_i^{(1)}) := NN_1(h_i^{(1)})$$

$$g(h(\mathbf{s}_1), h(\mathbf{s}_2), ..., h(\mathbf{s}_O)) := NN_2(\sigma(\sum_{i \in O} h_k^{(2)}(h_i^{(1)}))),$$

where NN_1 is a mapping from $\mathbb{R}^I$ to $\mathbb{R}^K$ defined by a neural network, and NN_2 is a mapping from $\mathbb{R}^K$ to $\mathbb{R}^{2H}$ defined by another neural network.

B.2 Approximation Difficulty of the Acquisition Function

Traditional variational approximation approaches provide wrong approximation direction when applied in this case (resulting in an upper bound of the objective $R_\phi(i, \mathbf{x}_O)$ which we maximize). Justification issues aside, (black box) variational approximation requires sampling from approximate posterior $q(\mathbf{z}|\mathbf{x}_O)$, which leads to extra uncertainties and computations. For common proposals of approximation:

- Directly estimate entropy via sampling ⇒ problematic for high dimensional target variables
- Using reversed information reward $\mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)}[D_{KL}(p(\mathbf{x}_\phi|\mathbf{x}_o)||p(\mathbf{x}_\phi|\mathbf{x}_o, \mathbf{x}_i))]$, and then apply ELBO (KL-divergence) ⇒ This does not make sense mathematically, since this will result in upper bound approximation of the (reversed) information objective, this is in the wrong direction.
- Ranganath's bound (Ranganath et al., 2016) on estimating entropy⇒ gives upper bound of the objective, wrong direction.
- All the above methods also needs samples from latent space (therefore second level approximation needed).

B.3 Connection of EDDI information reward with BALD

We briefly discuss connection of EDDI information reward with BALD (Houlsby et al., 2011) and. MacKay's work (MacKay, 1992). Assuming the model is correct, i.e. $q = p$, we have

$$\begin{aligned} R(i, \mathbf{x}_o) = {} & \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \left[D_{KL}(p(\mathbf{z}|\mathbf{x}_i, \mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_o))\right] \\ & - \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i, \mathbf{x}_o)} \left[D_{KL}(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right]. \end{aligned}$$

Note that based on McKay's relationship between entropy and KL-divergence reduction, we have:

$$\begin{aligned} & \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \left[D_{KL}(p(\mathbf{z}|\mathbf{x}_i, \mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_o))\right] \\ = & \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_i, \mathbf{x}_o)) - H(p(\mathbf{z}|\mathbf{x}_o))]\right]. \end{aligned}$$

Similarly, we have

$$\begin{aligned} & \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i, \mathbf{x}_o)} \left[D_{KL}(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right] \\ = & \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_\phi, \mathbf{x}_o)} \left[D_{KL}(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right] \\ = & \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_\phi, \mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o)) - H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right] \\ = & \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i, \mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o))\right] - \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_\phi, \mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right] \\ = & \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i, \mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o))\right] - \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right], \end{aligned}$$

where MacKay's result is applied to $\mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_\phi, \mathbf{x}_o)} \left[D_{KL}(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right]$. Putting everything together, we have

$$\begin{aligned} R(i, \mathbf{x}_o) = {} & \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_i, \mathbf{x}_o)) - H(p(\mathbf{z}|\mathbf{x}_o))]\right] \\ & - \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i, \mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o))\right] + \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right] \\ & = \left\{\mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_i, \mathbf{x}_o))\right] - \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i, \mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_i, \mathbf{x}_o))\right]\right\} \\ & - \left\{\mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_o))\right] - \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_o)} \left[H(p(\mathbf{z}|\mathbf{x}_\phi, \mathbf{x}_o))\right]\right\}. \end{aligned}$$

We can show that

$$
\begin{aligned}
&H(p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)) - \mathbb{E}_{\mathbf{x}_\phi \sim p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)}\left[H(p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o))\right] \\
=&-\int_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)\log p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)d\mathbf{z} + \int_{\mathbf{z},\mathbf{x}_\phi} p(\mathbf{z},\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)\log p(\mathbf{z}|\mathbf{x}_\phi,\mathbf{x}_i,\mathbf{x}_o) \\
=&\int_{\mathbf{z},\mathbf{x}_\phi} p(\mathbf{z},\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)\log \frac{p(\mathbf{z},\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)}{p(\mathbf{z}|\mathbf{x}_i,\mathbf{x}_o)p(\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o)} \\
=&\mathscr{I}\left[\mathbf{z},\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o\right],
\end{aligned}
$$

which is exactly the conditional mutual information $\mathscr{I}\left[\mathbf{z},\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o\right]$ used in BALD. Therefore, our chain rule representation of reward function leads us to

$$
R(i,\mathbf{x}_o) = \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)}\mathscr{I}\left[\mathbf{z},\mathbf{x}_\phi|\mathbf{x}_i,\mathbf{x}_o\right] - \mathbb{E}_{\mathbf{x}_i \sim p(\mathbf{x}_i|\mathbf{x}_o)}\mathscr{I}\left[\mathbf{z},\mathbf{x}_\phi|\mathbf{x}_o\right].
$$