# Sequential Learning and Regularization in Variational Recurrent Autoencoder

Jen-Tzung Chien
*Department of Electrical and Computer Engineering*
*National Chiao Tung University*, Hsinchu, Taiwan

Chih-Jung Tsai
*Department of Electrical and Computer Engineering*
*National Chiao Tung University*, Hsinchu, Taiwan

***Abstract*—Latent variable model based on variational autoencoder (VAE) is influential in machine learning for signal processing. VAE basically suffers from the issue of posterior collapse in sequential learning procedure where the variational posterior easily collapses to a prior as standard Gaussian. Latent semantics are then neglected in optimization process. The recurrent decoder therefore generates noninformative or repeated sequence data. To capture sufficient latent semantics from sequence data, this study simultaneously fulfills an amortized regularization for encoder, extends a Gaussian mixture prior for latent variable, and runs a skip connection for decoder. The noise robust prior, learned from the amortized encoder, is likely aware of temporal features. A variational prior based on the amortized mixture density is formulated in implementation of variational recurrent autoencoder for sequence reconstruction and representation. Owing to skip connection, the sequence samples are continuously predicted in decoder with contextual precision at each time step. Experiments on language model and sentiment classification show that the proposed method mitigates the issue of posterior collapse and learns the meaningful latent features to improve the inference and generation for semantic representation.**

***Index Terms*—sequential learning, Bayesian learning, recurrent neural network, variational autoencoder, language model**

## I. Introduction

Generative neural networks have been emerging in the era of signal processing and machine learning. Generative adversarial network [1], [2], variational autoencoder (VAE) [3], [4], autoregressive neural network and normalizing flow [5], [6] have been extensively developed for a variety of applications [7]–[10]. Among these models, VAE has the advantages of utilizing the latent variables in model construction and characterizing the randomness in learning representation. By applying the variational inference in neural latent variable model, VAE is learned to estimate the distribution of latent variables [11] or equivalently infer the structure of disentangled features [12] from input samples. VAE is composed of an encoder as inference model to recognize the latent variable and a decoder as generative model to reconstruct the random signal. The encoder obtains latent representation corresponding to input sample while the decoder generates the synthesized data given by latent samples. Encoder and decoder in a latent variable model are jointly optimized by maximizing the evidence lower bound (ELBO) of log likelihood.

Although VAE has been successfully developed, we still face challenges in learning procedure for sequence data. Previous studies [13]–[15] showed that vanilla VAE could not generate meaningful sentences. The distribution of latent variable was reduced to a standard Gaussian. The generated samples were deficient in diversity. This phenomenon is undesirable since the variational posterior lacks dependence on input data. The issue of *posterior collapse* happens because the Kullback-Leibler (KL) divergence between variational posterior and standard Gaussian prior in ELBO is close to zero. The variational posterior barely learns temporal information from input time-series signals. This causes a meaningless latent representation. Sampling from this posterior has no difference from sampling by a standard Gaussian. VAE is then realized as an autoregressive and generative model where the underlying structure of data was disregarded. To deal with this challenge, the von Mises-Fisher distribution was used to replace Gaussian distribution in a latent variable model [16]. Also, the one-dimensional convolutional neural network [14] was exploited as a hierarchical decoder to tackle this challenge by restricting the receptive field in a temporal convolutional network.

This paper proposes a new Bayesian framework to deal with the dilemma in variational sequential learning [17], [18] which is caused by collapse of KL term in ELBO during optimization procedure. By referring [19], [20], this work is motivated to learn an informative prior by using the Gaussian mixture model which encourages a flexible construction of latent space from training data. A variational mixture prior is learned. In the implementation, the encoder and decoder are further strengthened by performing the amortized regularization and skip connection, respectively. Amortized regularization leads to a smooth encoder for sequence data. This smooth encoder compresses the neighboring sequences from observation space into nearby locations in latent space. Owing to the preservation of distribution of temporal structure, the latent code is embedded with semantic and stochastic meaning in sequence data. Moreover, the skip connection from latent code to hidden state in recurrent network is enforced at each time so as to enrich latent information for prediction of output sequence. Information loss is reduced during propagation of recurrent steps in decoding which leads to a desirable latent space.

## II. Variational Recurrent Autoencoder

Variational neural models based on variational autoencoder (VAE) and variational recurrent autoencoder (VRAE) are first introduced. Basically, VAE is a neural machine consisting of an encoder and a decoder where the encoder acts as an inference model for latent distribution and the decoder serves as a generative model for reconstructing signals from latent distribution. VAE is learned from a set of training signals $\mathbf{x}$ by optimizing over a model with decoder parameter $\theta$ and latent variable $\mathbf{z}$. In learning procedure, VAE introduces a variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ with encoder parameter $\phi$ to approximate true posterior $p(\mathbf{z}|\mathbf{x})$. The evidence lower bound (ELBO) of log marginal likelihood is formulated as a learning objective for maximization which is decomposed in a form of

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})) \quad (1)$$

where $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a standard Gaussian, and the first and second terms in ELBO denote the reconstruction and regularization losses, respectively. The posterior collapse happens when the regularization loss or KL term goes to zero. In [6], [8], [21], the normalizing flow was proposed to obtain flexible variational distribution. Normalizing flow utilized a number of *invertible* transformations to assure richness and smoothness in latent distribution. In [13], [22], VRAE was proposed for sequence generation where two individual recurrent neural networks (RNNs) were employed in encoder and decoder. VRAE was developed for modeling of music signals and text data. As illustrated by the recursive nature of VRAE in Figure 1(a), the encoder RNN recursively characterizes time samples $\mathbf{x} = \{\mathbf{x}_t\}_{t=1}^T$ and finally infers random variable $\mathbf{z}$ by using the deterministic hidden state at last time $\mathbf{h}_T$. Meanwhile, the decoder RNN reconstructs the input sequence $\mathbf{x}$ recursively from the initial hidden state $\mathbf{s}_0$ and the begin of sentence <BOS>. The initial hidden state $\mathbf{s}_0$ is obtained by a mapping or embedding function based on the first input <BOS> and the latent code $\mathbf{z}$ sampled from the variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$. In practice, the teacher forcing is imposed on decoder to prevent wrong predictions recursively affecting later predictions during training procedure. In [15], [23], [24], the posterior collapse in VAE or VRAE was compensated by adjustable hyperparameter, strong encoder or self attention.

## III. Sequential Learning and Regularization

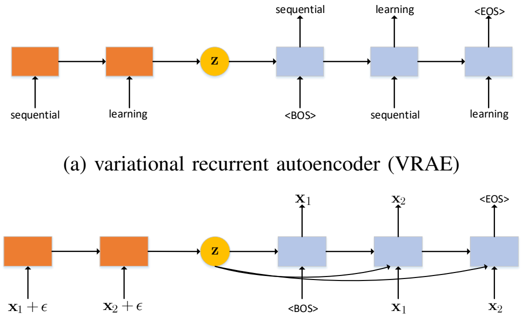
To deal with the difficulties in Bayesian sequential processing and learning, we integrate the variational mixture prior, the amortized regularization and the skip connection for robust sequence representation. This method tackles the issue of posterior collapse through different components in VRAE ranging from *encoder* to *decoder* and *latent variable*.

### A. Amortized Regularization on Encoder

Traditionally, variational inference is implemented by optimizing $p(\mathbf{x})$ using individual samples $\mathbf{x} = \{\mathbf{x}_t\}$. This is usually impractical when a large dataset $\mathbf{x}$ is adopted. The amortized variational inference replaces the per-sample optimization over $p(\mathbf{x})$ by means of an inference model $q_\phi(\mathbf{z}|\mathbf{x})$ driven by encoder parameter $\phi$. In general, VAE relies on the amortized inference which accelerates the computation by amortizing the optimization on each sample. An inference model is then optimized by using all data samples. In addition to this acceleration, the amortized variational inference was treated as a regularization for maximum likelihood estimation [9]. An alternative learning objective to ELBO was derived as

$$\max_\theta \left( \mathbb{E}_{\widehat{p}(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] - \min_\phi \mathbb{E}_{\widehat{p}(\mathbf{x})}\left[D_{\text{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})\right)\right]\right) \quad (2)$$

where a uniform distribution $\widehat{p}(\mathbf{x})$ over a dataset $\mathbf{x}$ is used. The choice of variational parameter $\phi$ or amortized inference model $q_\phi(\mathbf{z}|\mathbf{x})$, via KL minimization, actually regularizes the optimization of marginal likelihood $p_\theta(\mathbf{x})$. This is implemented by injecting the isotropic Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, i.e. using $q_\phi(\mathbf{z}|\mathbf{x} + \boldsymbol{\epsilon})$ in Eq. (2) or using hidden states $\mathbf{h}_t + \boldsymbol{\epsilon}$ in RNN encoder. The *denoising VRAE* is constructed to regularize the mapping or control the smoothness of an inference model. Smoothness indicates that neighboring data samples are mapped to similar locations in latent space. This property is merged in VRAE so that the sequence embedding can preserve semantic information, or equivalently the words with similar meanings are embedded with similar mappings.



(a) variational recurrent autoencoder (VRAE)

(b) VRAE with the amortized regularization and skip connection

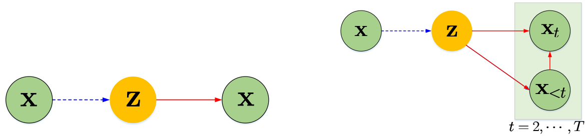
Fig. 1: Illustration of baseline and new sequence representations.

### B. Variational Mixture Prior for Latent Variable

Using VAE, the prior distribution $p(\mathbf{z})$ is often assumed as a standard Gaussian. Such a naive assumption likely leads to over regularization or posterior collapse in the estimated variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$, as reflected by KL term in Eq. (1). Inspired by [20], this study estimates the variational prior for VRAE by using $N_v$ *validation* sequences or sentences

$$p_\lambda(\mathbf{z}) = \frac{1}{N_v}\sum_{n=1}^{N_v} q_\phi(\mathbf{z}|\mathbf{x}_n) \quad (3)$$

which is seen as a sentence-level mixture prior of latent variable $\mathbf{z}$. The restriction of simple prior is then relaxed. ELBO is maximized to find variational prior $p_\lambda(\mathbf{z})$. However, this prior becomes infeasible if $N_v$ is large. In [20], the pseudo inputs were introduced as additional parameters to learn the amortized prior. Nevertheless, pseudo inputs could not be used in sequential learning since the length of each sequence was not fixed. In the experiments, we estimate the amortized mixture prior by using validation sentences, which is efficient and memory saving. The variational mixture prior is jointly learned with the variational posterior through stochastic backpropagation given by a shared $\phi$. A learnable prior leads to a multimodal and flexible latent space without increasing the number of parameters. A new variant of VRAE can be trained in an end-to-end style where the posterior collapse is alleviated in Bayesian sequential learning.



(a) VAE (b) VRAE with skip regularization

Fig. 2: Information paths. Dashed line denotes encoder with sampling.

### C. Skip Regularization on Decoder

Skip connection has been widely used in deep neural networks such as the residual network [25] or highway network [26]. In [27], skip connection was incorporated into VAE where the mutual information between observations $\mathbf{x}$ and latent codes $\mathbf{z}$ was maximized. Different from previous works, this study deals with sequence representation and carries out the skip connection on RNN decoder which is combined with the RNN encoder imposed by the amortized regularization and driven by the variational mixture prior for latent variable $\mathbf{z}$. An overall architecture is illustrated in Figure 1(b). RNN decoder basically predicts new sample $\mathbf{x}_t$ given by the previous inputs $\mathbf{x}_{<t}$. Information flow is passed from encoder to decoder only through the initial hidden state $\mathbf{s}_0$. However, in training stage of VRAE, the usage of teacher forcing provides another source of information. Such a source information reduces the dependence on the latent space, and accordingly causes the issue of posterior collapse. With the skip connection, the latent code $\mathbf{z}$ is enforced to join every prediction at different time steps $t$. The information path from encoder to the prediction is shortened. Figure 2 compares the information paths in VAE and VRAE. This comparison depicts how skip connection changes the generation process. This change mitigates the posterior collapse due to the restriction of RNN and the nature of autoregression. RNN is restricted by gradient vanishing and exploding. Autoregressive model is hard to make right prediction if past predictions are wrong. Skip connection copes with these issues in sequential learning from long sequences.

We construct a VRAE variant with amortized regularization, mixture prior and skip connection which is denoted by VRAE-AMS. The learning criterion $\mathcal{L}(\mathbf{x};\theta,\phi)$ is expressed by

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x}+\boldsymbol{\epsilon}) \middle\| \frac{1}{N_v}\sum_{n=1}^{N_v} q_\phi\left(\mathbf{z}|\mathbf{x}_n\right)\right). \tag{4}$$

The stochastic backpropagation procedure is addressed in Algorithm 1. Encoder function $f_\phi^{\mathrm{enc}}(\cdot)$ and decoder function $f_\theta^{\mathrm{dec}}(\cdot)$ calculate the hidden states $\mathbf{h}$ and $\mathbf{s}$ using recurrent inputs $\{\mathbf{x}_t, \mathbf{h}_{t-1}\}$ and $\{\mathbf{x}_t, \mathbf{s}_{t-1}, \mathbf{z}\}$, respectively. Skip connection and amortized inference are implemented. $f_\phi^{(q)}(\cdot)$ and $f_\theta^{(o)}(\cdot)$ calculate the variational and generative distributions for latent samples $\mathbf{z}$ and synthesized data $\mathbf{x}_t$, respectively. The reparameterization trick [3] is applied to draw random samples $\mathbf{z}$. Stochastic backpropagation via gradient ascent using Eq. (4) is performed for sequence generation and representation.

**Algorithm 1:** Training procedure for VRAE-AMS

```
Input minibatches of $\mathbf{x}$ & hyperparameter $\sigma^2$
Initialize parameters $\theta$, $\phi$ & hidden states $\mathbf{s}_0$, $\mathbf{h}_0$
for number of iterations do
    Encoder:
    for $t = 1, \cdots, T$ do
        $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$
        $\mathbf{h}_{t-1} \leftarrow \mathbf{h}_{t-1} + \boldsymbol{\epsilon}$
        $\mathbf{h}_t \leftarrow f_\phi^{\mathrm{enc}}(\mathbf{x}_t, \mathbf{h}_{t-1})$
    end
    $q_\phi(\mathbf{z}|\mathbf{x}) \leftarrow f_\phi^{(q)}(\mathbf{h}_T)$
    $\mathbf{z} \leftarrow$ sample from $q_\phi(\mathbf{z}|\mathbf{x})$
    Decoder:
    for $t = 1, \cdots, T$ do
        $\mathbf{s}_t \leftarrow f_\theta^{\mathrm{dec}}(\mathbf{x}_t, \mathbf{s}_{t-1}, \mathbf{z})$
        $p_\theta(\mathbf{x}_{t+1}|\mathbf{x}_{\leq t}, \mathbf{z}) \leftarrow f_\theta^{(o)}(\mathbf{s}_t)$
    end
    Compute learning objective $\mathcal{L}(\mathbf{x};\theta,\phi)$ in Eq. (4)
    Update parameters $\{\theta, \phi\}$ via gradient ascent by using
      $\{\nabla_\theta\mathcal{L}, \nabla_\phi\mathcal{L}\}$
end
```

## IV. Experiments

The performance of sequential learning were evaluated by semantic representation using three benchmark datasets: Penn TreeBank (PTB) [28], Yelp 2013 (Yelp) [16] and IMDB [29].

### A. Experimental Setup

In the experiments, the vanilla VRAE was seen as baseline system. VRAE using the normalizing flow (denoted by VRAE-F) [8], [21] was implemented. The inverse autoregressive flow was used to transform the posterior. There were ten convex combinations for flow transformation. For ablation study, VRAE-AS and VRAE-AM are implemented to evaluate the simplified variants without mixture prior and skip connection, respectively, VRAE-A, VRAE-M and VRAE-S were carried out by simplifying VRAE-AMS where only the amortized regularization [9], the variational mixture prior [20] and the skip connection [27] were performed, respectively. Annealing in $\beta$-VAE [23] was applied in different methods. Different from [8], [9], [20], [21], [27] based on VAE, this paper conducted sequential learning by using sequence data based on VRAE. The recurrent machine RNN was based on the long short-term memory (LSTM) [30]. The latent variable $\mathbf{z}$ was learned for sentence generation in language modeling [31], [32] as well as in sentiment classification. There were four metrics in language modeling tasks where negative log-likelihood (NLL) and perplexity (PPL) showed the ability of word generation and prediction, KL value reflected if the model prevented the posterior collapse, and the number of active units (AU) investigated how well the inference model was active to work. Larger KL indicated more likely the latent space $\mathbf{z}$ was learned. While we had 32 dimensions in latent space of $\mathbf{z}$, the models exploiting larger dimensions or active units were considered to work better in inference procedure. An active dimension was defined to achieve its variance to be greater than 0.01. A small set (5%) of collected data were held out as the validation sentences $\{\mathbf{x}_n\}_{n=1}^{N_v}$ to determine the variational mixture prior in Eq. (3). 85% of collected data was used as training data and the remaining 10% was adopted as test data. The hyperparameter $\sigma^2$ for amortized regularization was selected as 0.5. $t$-SNE [33] was used to visualize two-dimensional (2-D) samples from $\mathbf{z} \in \mathbb{R}^{32}$. Adam optimizer [34] with initial learning rate 0.001 was adopted. Gradient clipping was applied with maximum norm 5. Minibatch size was 32. Computation cost was evaluated.

### B. Language Modeling on Penn TreeBank

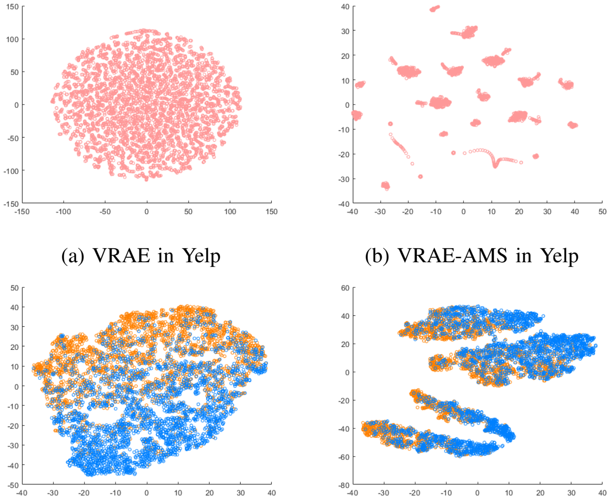
PTB is a standard dataset for evaluation of language model which predicts the next word based on the history words. In PTB, the average length of a sentence was 21.1 words. Vocabulary size was set to 8K. The proposed VRAE-AMS is assessed by different metrics. Table I reports the results of LSTM and different variants of VRAE-AMS. Overall, VRAE-AMS obtains the best performance for generation, as it achieves the lowest NLL and PPL. It successfully prevents posterior collapse and has the most effective result with the largest KL value. For a 32 dimensional latent space, VRAE-M exploits the largest dimensions among different VRAEs. VRAE-F utilizes 27 of them. Advanced VRAEs obtain much larger AU than vanilla VRAE. VRAE-S performs better than the other stand-alone methods. Skip connection is crucial.

TABLE I: Evaluation of LSTM and VRAEs for language modeling using PTB. The abbreviation 'F' means flow posterior, 'A' means amortized regularization, 'M' means mixture prior and 'S' means skip connection. The best numbers are bold.

| Model | NLL | KL | PPL | AU |
|---|---|---|---|---|
| LSTM | 102.27 | – | 132.89 | – |
| VRAE | 101.45 | 4.86 | 127.78 | 4 |
| VRAE-F | 101.24 | 4.79 | 122.35 | 27 |
| VRAE-A | 100.31 | 4.95 | 115.41 | 9 |
| VRAE-M | 100.88 | 4.97 | 117.03 | **29** |
| VRAE-S | 99.15 | 6.37 | 110.67 | 18 |
| VRAE-AS | 98.15 | 6.47 | 109.29 | 19 |
| VRAE-AM | 98.37 | 5.98 | 109.90 | 20 |
| VRAE-AMS | **97.69** | **6.58** | **106.81** | 25 |

### C. Language Modeling on Yelp 2013

Yelp is a restaurant review dataset collected from Yelp Dataset Challenge in year 2013. There were 47.6 words in average in a review. Vocabulary size was 12K. Results are shown in Table II. The training time of different VRAEs relative to vanilla VRAE was reported. 20 epochs were run. VRAE-AMS achieves the lowest value on NLL and PPL. The highest value in KL divergence was measured by VRAE-AS where there were 20 active units in 32 dimensions which was smaller than 26 of using VRAE-M. VRAE-AMS improves VRAE and outperforms the other models in generation performance. Comparing different standard alone components, the mixture prior spends the highest computation and the skip connection performs the best in NLL and PPL. Nevertheless, the complementary processing and learning on encoder, decoder and latent distribution do improve system performance. In addition, Figures 3(a) and (b) compare the global structures of 2-D latent space of VRAE and AMP-VRAE, respectively. VRAE has a latent space in shape of a circle, which indicates Gaussian distribution with diagonal covariances. VRAE-AMS, on the other hand, reflects the multi-modal mixture distribution, which leads to rich information in latent space.



(a) VRAE in Yelp

(b) VRAE-AMS in Yelp

(c) VRAE in IMDB

(d) VRAE-AMS in IMDB

Fig. 3: Latent distributions on Yelp and IMDB datasets.

TABLE II: Evaluation of different methods using Yelp.

| Model | NLL | KL | PPL | AU | Time |
|---|---|---|---|---|---|
| LSTM | 196.69 | – | 62.91 | – | – |
| VRAE | 196.28 | 2.25 | 62.38 | 3 | 1x |
| VRAE-F | 195.83 | 2.21 | 61.21 | 19 | 1.02x |
| VRAE-A | 194.98 | 2.35 | 59.19 | 8 | 2.06x |
| VRAE-M | 194.35 | 2.40 | 59.85 | 24 | 2.69x |
| VRAE-S | 192.39 | 4.99 | 57.37 | 22 | 2.17x |
| VRAE-AS | 192.11 | **5.19** | 57.11 | 20 | 3.17x |
| VRAE-AM | 192.24 | 5.09 | 57.39 | 23 | 3.25x |
| VRAE-AMS | **191.80** | 5.18 | **56.76** | **26** | 3.69x |

### D. Sentiment Classification on IMDB

IMDB is the movie review dataset collected from the Internet Movie Database website. IMDB contained 50K labeled data with even number of positive and negative reviews. The average length in a review was 78.2 words. Vocabulary size was 20K. Evaluations on language modeling and sentiment classification were performed. In the implementation, an additional classifier was connected to the learned latent samples $\mathbf{z}$ and used to predict if the review is positive or negative. The classifier was jointly trained with the sequence

TABLE III: Evaluation of different methods using IMDB.

| Model | NLL | KL | PPL | AU | Accu(%) |
|---|---|---|---|---|---|
| VRAE | 387.67 | 1.56 | 141.58 | 4 | 68.04 |
| VRAE-F | 387.86 | 1.85 | 141.92 | **25** | 67.94 |
| VRAE-A | 386.37 | 2.06 | 139.24 | 10 | 69.94 |
| VRAE-M | 386.69 | 2.09 | 139.82 | 9 | 68.76 |
| VRAE-S | 383.25 | 2.32 | 137.23 | 20 | 70.10 |
| VRAE-AS | 382.84 | 2.29 | 137.11 | 22 | 71.22 |
| VRAE-AM | 383.02 | 2.25 | 138.05 | 18 | 70.85 |
| VRAE-AMS | **381.92** | **2.92** | **136.23** | 24 | **72.30** |

representation using VRAE-AMS. The NLL, KL, PPL, AU and classification accuracy were reported. Table III shows that VRAE-AMS achieves the best results in most metrics. Although the active units are less than those in VRAE-F, the accuracy using VRAE-AMS is still improved. Figures 3(c) and (d) also show the 2-D latent spaces of VRAE and VRAE-AMS, respectively. Orange indicates the positive reviews while blue indicates the negative reviews. VRAE and VRAE-AMS can separate most of reviews in different classes. Source codes of VRAE-AMS are implemented by PyTorch and accessible at https://github.com/NCTUMLlab/Chih-Jung-Tsai/.

## V. Conclusions

This paper have presented the Bayesian sequential learning for semantic representation which was applied for language modeling as well as sentiment classification. A new variational recurrent autoencoder was proposed by incorporating the amortized regularization, the variational mixture prior and the skip connection for regularized processing and learning in encoder, latent space and decoder. Amortized regularization resulted in smoothing the encoder and extracting the semantic information. Variational mixture prior led to rich latent variable representation. Skip connection reinforced the latent code to join each prediction in the decoder. Experimental results on three tasks showed that these three complementary schemes alleviated the issue of posterior collapse and improved the performance of VRAE for sequence representation.

## References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[2] J.-T. Chien and C.-L. Kuo, "Variational Bayesian GAN," in *Proc. of European Signal Processing Conference*, 2019, pp. 1454–1458.

[3] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[4] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. of International Conference on Machine Learning*, 2014, pp. 1278–1286.

[5] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[6] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows," *arXiv preprint arXiv:1505.05770*, 2015.

[7] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10215–10224.

[8] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Advances in Neural Information Processing Systems*, 2016.

[9] R. Shu, H. H. Bui, S. Zhao, M. J. Kochenderfer, and S. Ermon, "Amortized inference regularization," in *Advances in Neural Information Processing Systems*, 2018, pp. 4393–4402.

[10] J.-T. Chien and Y.-T. Bao, "Tensor-factorized neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1998–2011, 2018.

[11] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational lossy autoencoder," in *Proc. of International Conference on Learning Representations*, 2017.

[12] E. Mathieu, T. Rainforth, S. Narayanaswamy, and Y. W. Teh, "Disentangling disentanglement," *arXiv preprint arXiv:1812.02833*, 2018.

[13] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. of SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 10–21.

[14] S. Semeniuta, A. Severyn, and E. Barth, "A hybrid convolutional variational autoencoder for text generation," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 627–637.

[15] J.-T. Chien and C.-W. Wang, "Variational and hierarchical recurrent autoencoder," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3202–3206.

[16] J. Xu and G. Durrett, "Spherical latent spaces for stable variational autoencoders," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2018.

[17] J.-T. Chien and Y.-C. Ku, "Bayesian recurrent neural network for language modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 361–374, 2016.

[18] J.-T. Chien, "Deep Bayesian natural language processing," in *Proc. of Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, 2019, pp. 25–30.

[19] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical variational auto-encoders," *arXiv preprint arXiv:1804.00891*, 2018.

[20] J. M. Tomczak and M. Welling, "VAE with a VampPrior," *arXiv preprint arXiv:1705.07120*, 2017.

[21] J. M. Tomczak and M. Welling, "Improving variational auto-encoders using convex combination linear inverse autoregressive flow," *arXiv preprint arXiv:1706.02326*, 2017.

[22] O. Fabius and J. R. van Amersfoort, "Variational recurrent auto-encoders," *arXiv preprint arXiv:1412.6581*, 2014.

[23] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "$\beta$-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. of International Conference on Learning Representations*, 2017.

[24] J.-T. Chien and C.-W. Wang, "Self attention in variational sequential learning for summarization," *Proc. of Annual Conference of International Speech Communication Association*, pp. 1318–1322, 2019.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[26] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[27] A. B. Dieng, Y. Kim, A. M. Rush, and D. M. Blei, "Avoiding latent variable collapse with generative skip models," *arXiv preprint arXiv:1807.04863*, 2018.

[28] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," in *Proc. of IEEE Spoken Language Technology Workshop*, 2012, pp. 234–239.

[29] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. of Annual Meeting of Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2011, vol. 1, pp. 142–150.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] J.-T. Chien, "Association pattern language modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1719–1728, 2006.

[32] J.-T. Chien and C.-H. Chueh, "Joint acoustic and language modeling for speech recognition," *Speech Communication*, vol. 52, no. 3, 2010.

[33] L. van der Maaten and G. E. Hinton, "Visualizing data using *t*-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.