Empirical Prior for Text Autoencoders

Yongjing Yin[1,2], **Wenyang Gao**[2], **Haodong Wu**[2], **Jianhao Yan**[2], **Yue Zhang**[2,3*]
[1]Zhejiang University
[2]School of Engineering, Westlake University
[3]Institute of Advanced Technology, Westlake Institute for Advanced Study
{yinyongjing,gaowenyang,wuhaodong,yanjianhao}@westlake.edu.cn
yue.zhang@wias.org.cn

Abstract

This paper explores the application of Variational Autoencoders (VAE) in text generation, focusing on overcoming challenges like posterior collapse and the limitations of simplistic prior distributions. We investigate a transition from VAE to text autoencoders (AE), which model a compact latent space and preserve the capability of the language model itself. Our method involves layer-wise latent vectors regularized by orthogonal constraints to encourage distinct semantic spaces. In particular, we estimate an empirical prior online from the learned latent vectors to support sampling during generation like VAE. Experimental results on standard benchmarks demonstrate that the autoencoders generate higher quality and more diverse text than the state-of-the-art VAE-based Transformer models, offering an effective alternative for generative language modeling.

1 Introduction

Variational Autoencoder (VAE) offers an effective approach to train generative models with latent variables (Kingma and Welling, 2014; Rezende et al., 2014). By adopting the paradigm for training language models, the latent variables can help to capture the underlying causal structure of the generative process more effectively, and provide an interpretable representation of high-level features like topics or syntactic properties (Bowman et al., 2016; Hu et al., 2017; Hu and Li, 2021; Hu et al., 2022). Moreover, representing sentences in a low-dimensional latent space facilitates manipulation and guided generation using interpretable vector operators. VAEs have demonstrated their success in generating stylistic text (John et al., 2019; Hu and Li, 2021), stories (Yu et al., 2020; Fang et al., 2021), and dialog (Yang et al., 2023). Optimus (Li et al., 2020), the pioneering large-scale pretrained VAE for text, underscoring the advantages highlighted during the pre-training phase.

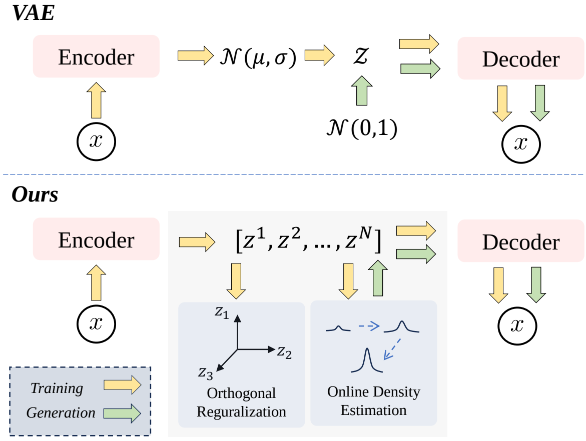


Figure 1: Illustration of VAE and AE-based models.

While VAEs possess promising theoretical advantages, they face challenges such as posterior collapse (Bowman et al., 2016; van den Oord et al., 2017; Dai and Wipf, 2019), Successfully training these models necessitates the use of carefully designed techniques (Bowman et al., 2016; Higgins et al., 2017; Yang et al., 2017; He et al., 2019; Fu et al., 2019), and the learned posterior often deviates significantly from the hypothetical prior, leading to undermining the quality of the generated text. The drawbacks of VAE can be attributed to the simplistic prior and its restriction on the latent distribution induced by the KL divergence term (Dai and Wipf, 2019) (the upper part of Figure 1). Following the practice of VAE in image generation, text VAE adopts such a prior to represent plausible language. However, autoregressive language models themselves possess a strong ability to generate linguistically valid outputs, which weaken the role of the hypothetical prior (Bowman et al., 2016; Yang et al., 2017; Hu et al., 2022).

Based on the above observations, we investigate the potential of AE in achieving what can be achieved by text VAE. Typically, VAE degenerates to AE when removing the regularization of samples to the prior. The text AE precisely preserves the

[*] Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 13628–13640
November 12-16, 2024 ©2024 Association for Computational Linguistics

semantics of the input but lacks the capability of generative modeling, specifically the ability to sample novel latent vectors. To address this issue, we propose to estimate the prior based on the learned features of AE, i.e., empirical prior distribution. The whole model is illustrated in the lower half of Figure 1. Specifically, we introduce multiple latent vectors for AE induced from different encoder layers. To encourage the development of expressive and distinct semantic space, we regularize the latent vectors with an orthogonal regularization. The empirical prior is derived by summarizing the distribution of the latent vectors from the training data while respecting the modeling capability of the language model itself, and we employ an online method to estimate the empirical prior for efficiency.

We conduct experiments on the standard benchmarks used to evaluate text VAE in terms of reconstruction perplexity and evaluation of generated samples. Whether using deterministic latent vectors or sampling the latent vectors from the empirical posterior, the perplexity of the AE is lower than that of the VAE models. The underlying reason is that not adding a simplistic prior avoids interfering with the language model's own capability, greatly increasing fidelity. In terms of generation, the text generated using sampling from the empirical prior distribution outperforms the state-of-the-art VAE models (e.g., DELLA (Hu et al., 2022)) on quality and diversity. The analyses demonstrate that the AE model requires fewer latent variables, and the visualization shows that the AE model learns a similar latent space structure to DELLA. We will release the code upon acceptance.

2 Related Work

VAE In image VAE, a simplistic prior often leads to posterior collapse, or, in cases where collapse is avoided, a pronounced mismatch between the prior and posterior distributions can compromise generative performance. Consequently, researchers have explored nuanced priors, necessitating corresponding adjustments to model architectures and training objectives (Kingma et al., 2016; Tomczak and Welling, 2018; Dai and Wipf, 2019; Vahdat and Kautz, 2020). Ghosh et al. (2020) investigate the potential of regularized autoencoders on small image generation datasets, which use Gaussian Mixture Models to conduct ex-post density estimation.

In NLP, text VAE leverages various intriguing characteristics of the latent space (John et al., 2019; Li et al., 2020; Hu and Li, 2021). However, the modeling capacity and empirical performance are limited primarily due to KL vanishing. In such cases, the decoder disregards the latent space entirely and degenerates into a simplified language model. Various training techniques have been proposed such as annealing (Bowman et al., 2016; He et al., 2019; Li et al., 2019; Fu et al., 2019), KL thresholding (Zhu et al., 2020), and the combination with pretrained Transformer models (Li et al., 2020; Hu et al., 2022). Li et al. (2019) pretrain the inference network with an autoencoder objective. In addition, there have been efforts to propose more powerful prior distributions (Pelsmaeker and Aziz, 2020; Ding and Gimpel, 2021; Dai et al., 2021; Fang et al., 2022; Yang et al., 2023). For example, DPrior (Fang et al., 2022) uses learnable vectors (dictionary atoms) to introduce a data-driven prior, and Dior-CVAE (Yang et al., 2023) employs a diffusion model to enhance the prior.

Nevertheless, there is a lack of research addressing the necessity of the prior distribution in this context. Considering the inherent robustness of language models in effectively modeling and generating coherent text, allowing the prior distribution to adapt to the empirical language distribution may be more advantageous than enforcing the condition variables to conform to a fixed prior distribution. We are the first to demonstrate that removing the hardcoded prior makes the AE-based language models achieve better sample quality and diversity performance than the state-of-the-art VAE Transformers.

Li et al. (2020) first pretrain a big language model in the VAE paradigm, and indicate its superior controllability and capability than casual language models. They strive to assist the research community in acknowledging the significance of latent conditional language modeling in pre-training and making it more feasible in practice. However, the training difficulty may limit the development of scaling up text VAE. In comparison, AE is naturally suitable for large-scale pretraining, which can be traced back to the restricted Boltzmann machines (Hinton and Salakhutdinov, 2006). The additional high-level guidance before the next token prediction can help to capture the underlying causal structure of the generative process. Moreover, the learned latent space allows better controllability and interpretability. These advantages can be more appealing when further scaling up.

Conditional Language Models. Conditional language models can generate sentences with desired attributes such as sentiments or topics. CTRL (Keskar et al., 2019) extends Transformer to a conditional one which introduces various control codes as prefixes. POINTER (Zhang et al., 2020) uses an insertion-based method for hard-constrained text generation. Similarly, CoCon (Chan et al., 2021) introduces a conditional control module into the GPT model. These conditions are symbolic and require feature engineering. Relying on such coarse features may limit models' capacity to comprehend and generate language effectively. Text VAE can be regarded as latent conditional language models by leveraging the expressive power of compact latent spaces (Bowman et al., 2016; Hu et al., 2017; Hu and Li, 2021). Similar to text VAE, we use low-dimensional latent vectors to represent conditions. Instead of restricting the latent to a predefined prior like in VAE, we investigate the feasibility of AE-based conditional language models with the introduction of an empirical prior.

3 Method

3.1 Neural Language Models

Given an observed text sequence of length T, $x = \{x_1, ..., x_T\}$, neural language models (Bengio, 2008) (NLM) are trained to generate every token conditioned on the previous tokens

$$p_\theta(x) = \prod_{t=1}^{T} p_\theta(x_t | x_{<t}), \tag{1}$$

where θ is the model parameter. The model is typically trained via maximum likelihood estimation, and the representative model family is GPT (Radford et al., 2019; Brown et al., 2020). The generation process relies solely on previous words, which limits their ability to be guided by higher-level structures, such as tense and topics.

3.2 VAE

Different from the conventional NLM, the decoder of VAE takes sampled latent variables as conditional factors for generation. Concretely, a latent vector z is first sampled from the prior distribution of the latent space $p(z)$, and the *decoder* generates the text sequence x from a conditional distribution $p_\theta(x|z)$ in an auto-regressive manner:

$$p_\theta(x|z) = \prod_{t=1}^{T} p_\theta(x_t | x_{<t}, z). \tag{2}$$

In particular, $p(z)$ is usually assumed to be a standard Gaussian distribution. Unlike the conventional language models which take a prefix as input, the VAE-based conditional language models use the latent vector z to determine the high-level semantics.

The training procedure of VAE is also known as inference. The parameter set of the decoder θ is typically learned by maximizing the marginal log-likelihood:

$$\log p_\theta(x) = \log \int p_\theta(x|z)p(z)dz. \tag{3}$$

Due to the intractable optimization, the variational inference is introduced and the objective is changed to maximize the evidence lower bound (ELBO):

$$\begin{aligned}\log p_\theta(x) \geq \mathcal{L}_{\text{ELBO}} =& \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right] - \\ & \text{KL}\left(q_\phi(z|x) || p(z)\right). \end{aligned} \tag{4}$$

In practice, VAE utilizes amortized variational inference (Kingma and Welling, 2014; Mnih and Gregor, 2014), which introduces a neural *encoder* $q_\phi(z|x)$ to approximate the true posterior. Moreover, $q_\phi(z|x)$ is usually assumed to be Gaussian and the re-parametrization trick can be used. The training process is illustrated in the upper part of Figure 1, the posterior is regularized to match the prior, and the decoder takes the latent vectors sampled from the posterior as the input.

3.3 AE with Empirical Prior Estimation

In this section, we describe the details of the AE model including the encoder, decoder, training procedure, and the method of empirical distribution estimation.

Encoder As shown in the lower half of Figure 1, we model layer-wise latent vectors $Z = \{z^i\}_{i=1}^N$, where N denotes the number of latent vectors. For each input sequence x, we append a virtual token s to absorb the semantic information of the input sequence. We feed the whole sequence $\{x_1, x_2, ..., x_T, s\}$ into the encoder, and obtain a sequence of contextualized representations of the virtual token across layers $\{s^i\}_{i=1}^N$. Each z^i is calculated by projecting s^i into a lower dimension:

$$z^i = W_{down}^i s^i + b_{down}^i, \tag{5}$$

where $W^i_{down} \in R^{d_h \times d_z}$ and b^i_{down} are the trainable parameters, and d_h and d_z indicate the dimension of the hidden states and latent vectors, respectively. The linear transformations are not shared considering distinct semantic spaces in different layers. In addition, we keep a latent vector for every certain layer to enable compact semantic representations, considering that the representations of adjacent layers are relatively similar (van Aken et al., 2019; Sajjad et al., 2022). We can easily adjust the number of latent vectors, and the maximum number of latent vectors equals the number of encoder layers. Compared to VAE, the encoder output of AE is deterministic latent features instead of the parameters of the posterior.

Decoder We treat Z as additional memory attended by other tokens via self-attention, which has shown the superiority to regarding Z as extra input token embeddings (Li et al., 2020). Specifically, for z^i obtained in the ith encoder layer, we only inject it into the ith decoder layer rather than all the layers, to encourage the latent vectors to learn different aspects of the input semantics. Before being attended, z^i is projected into the same dimension of hidden states of the decoder with a distinct linear transformation:

$$h^i_z = W^i_{up} z^i + b^i_{up}, \tag{6}$$

where $W^i_{up} \in R^{d_z \times d_h}$ and b^i_{up} are the trainable parameters. Such an infusion mechanism of latent vectors requires no modification of the decoder architecture.

Training The overall training objective is

$$\mathcal{L} = -\frac{1}{|x|} \sum_{i}^{|x|} \text{log} p_\theta(x_i|x_{<i}, Z) + \lambda \mathcal{L}_{ort}, \tag{7}$$

where λ is the coefficient of the additional orthogonal regularization $\mathcal{L}_{ort}$ on the latent vectors:

$$\mathcal{L}_{ort} = \frac{1}{K^2} \left\| \ell_2(Z)^\top \ell_2(Z) - I_K \right\|^2_2, \tag{8}$$

where $\ell_2(*)$ denotes L2 normalization on each latent vector and I_K denotes the identity matrix. The regularization is used to encourage the learned latent vectors as the basis in the latent space and reduce information redundancy.

Empirical Prior Estimation During training, we employ an efficient online method to estimate the empirical prior distribution for each z^i, based on all of the observed latent features encoded by the model. Specifically, we assume the distribution Gaussian $\mathcal{N}(\mu, \sigma^2\mathbf{I})$ and update the parameters with an exponential moving average (EMA):

$$\mu = \lambda \mu_B + (1-\lambda)\mu, \tag{9}$$

$$\sigma^2 = \lambda \sigma^2_B + (1-\lambda)\sigma^2, \tag{10}$$

$$\mu_B = \frac{1}{m} \sum_{i=1}^{m} z \tag{11}$$

$$\sigma^2_B = \frac{1}{m} \sum_{i=1}^{m} (z - \mu_B)^2, \tag{12}$$

where $\lambda \in [0, 1)$ is a momentum coefficient of EMA and is set to 0.1. Moreover, B denotes a mini-batch of size m, and μ_B and σ^2_B denote the empirical mean and variance of B, respectively. For better readability, we have omitted the superscript i in the formulas. When the training is finished, we can sample latent vectors from the estimated empirical prior and use them to generate new sentences, which is similar to the generation procedure of VAE. We can also use offline methods to estimate the prior, and we discuss the comparison in Section 4.4.

In addition, we can use the observed features in the training set based on nearest neighbor retrieval to estimate the empirical posterior $q(z^i|x)$ for each z^i. Concretely, we run the trained encoder by an additional forward pass over the training data and store the latent vectors. Then, we use each dimension z^i_j of the latent vector z^i as a pivot to retrieve M neighbor latent variables with L1 distance. The retrieved values comprise a set of new latent vectors $\{\hat{z}^m\}^M$, which are used to estimate the empirical posterior. If the $q(z^i|x)$ is assumed to be Gaussian, the retrieved latent vectors are used to calculate the empirical variance.

4 Experiments

4.1 Settings

We evaluate our model on the representative benchmarks for text VAE, i.e., Yelp (Yang et al., 2017), Yahoo (He et al., 2019), and SNLI (Bowman et al., 2015). The preprocessing follows (Hu et al., 2022), and the numbers of sentences of the training set, development set, and test set are 100K, 10K, and 10K, respectively. We focus on the Transformer VAE models as the baselines, and the setting follows DELLA (Hu et al., 2022) for a fair comparison.

Model	#P	PPL↓	AU↑	Quality		Diversity		
				BLEU↑	MAUVE↑	Self-BLEU↓	Dist↑	JS↓
				Yelp				
GPT-2	124M	22.13	-	56.92	0.12	65.90	17.96	0.51
Optimus	234M	22.79	32	-	-	-	-	-
Embedding	124M	19.98	6	56.34	0.42	65.27	15.59	0.44
Memory	125M	19.95	11	**57.37**	0.46	63.90	16.91	0.39
Softmax	124M	20.14	13	56.83	0.45	64.26	16.51	0.40
DELLA	193M	12.35	23	57.15	0.55	60.02	17.63	0.43
AE(EMA)	125M	**6.04**	**32**	54.14	0.61	**56.50**	**20.28**	**0.25**
AE(Full)				55.96	**0.69**	58.63	18.62	0.26
AE(Diag)				55.04	0.61	57.57	19.52	0.26
				Yahoo				
GPT-2	124M	24.17	-	44.25	0.15	54.06	21.07	0.28
Optimus	234M	23.11	32	-	-	-	-	-
Embedding	124M	22.18	3	42.27	0.31	54.15	20.80	0.32
Memory	125M	22.03	18	**45.20**	0.37	54.59	21.87	0.33
Softmax	124M	22.35	19	44.28	0.34	54.49	21.65	0.32
DELLA	193M	11.49	21	44.67	0.38	48.53	21.88	0.31
AE(EMA)	125M	**7.62**	**32**	42.11	**0.50**	**43.66**	**28.78**	0.26
AE(Full)				43.01	0.49	45.40	26.21	**0.27**
AE(Diag)				42.51	**0.50**	44.16	27.98	**0.27**

Table 1: Evaluation results on Yelp and Yahoo. The best results are highlighted in bold.

The encoder and decoder shared the same parameters initialized with 12-layer GPT-2 (Radford et al., 2019). The dimension of the latent variable is set as 32 for all of the models, and the number of latent vectors in AE is set as 4 and 3 for Yelp and Yahoo, respectively. The batch size is 128, and the learning rate is 5e-5. The dropout rate is set to 0.1 by default and set to 0.7 for SNLI. Beam search is adopted to generate sentences with a beam size of 10. In addition to EMA, we investigate offline methods of the empirical prior estimation and assume the latent variables with "Full" Gaussian and "Diag" Gaussian, which have the general covariance matrix and diagonal covariance matrix, respectively.

4.2 Baselines

We compare our models with the following baselines: GPT-2 (Radford et al., 2019), a fine-tuned language model; Optimus (Li et al., 2020), which is the first pretrained large text VAE with a pretrained BERT as the encoder and GPT-2 as the decoder; Embedding (Hu et al., 2022), in which the latent vector is added to the token embedding at each decoding step; Memory (Fang et al., 2021), in which the latent vector is attended by self-attention in each decoder layer; Softmax (Wang and Wan, 2019), which uses the latent vector to intervene the output softmax; and DELLA (Hu et al., 2022), which learns hierarchical latent variables with each inferred by each encoder layer and injected into the decoder layers by low-rank tensor product. Memory, Softmax, and Embedding are reimplemented by Hu et al. (2022) using the same setting.

4.3 Evaluation Metrics

The evaluation metrics include two parts. The first part measures the training performance of the generative language model including reconstruction perplexity (PPL) and active units (AU) (Burda et al., 2016) which denotes the total number of active units in z. The second part evaluates the quality and diversity of sentences generated based on latent variable sampling. Specifically, the quality metrics focus on the measurement of the divergence between human-written text and the generated one including BLEU (Papineni et al., 2002) and MAUVE (Pillutla et al., 2021). The diversity metrics measure the self-similarity of the generated sentences, and low-similarity sentences are preferred. We report Self-BLEU (Zhu et al., 2018), Dist (Li et al., 2016), and JS (Jaccard similarity) (Wang and Wan, 2018). The details of the metrics are shown in Appendix D.

4.4 Main Results

Table 1 presents the performance comparison between the VAE and the AE Transformers on Yelp and Yahoo.

First of all, the AE models achieve significantly lower PPL, which is highly intuitive. The low perplexity indicates the advantage of such a simpler training paradigm and a better fidelity to the input semantics through the latent conditions. Allowing the distribution of latent variables distribution to adapt to the learned features is more natural and powerful than forcing it to a predefined prior. Compared to the standard language model GPT-2, the advantage is more pronounced because reconstruction is easier when the semantics of the text to generate is known. Moreover, the AE models achieve higher AU scores, indicating the latent vectors are fully exploited to encode semantics.

One important role of text VAE is its generative ability to generate new text based on the abstract condition. The quality scores BLEU and MAUVE assess whether the distribution of the set of sentences generated based on the samples is faithful to the distribution of the test set. In this aspect, although the AE models obtain slightly lower BLEU scores, they achieve obviously higher scores on the model-based MAUVE, which demonstrates a much stronger correlation with human ratings (Pillutla et al., 2021). Concretely, the AE(EMA) models obtain 0.61/0.50 MAUVE scores on Yelp and Yahoo, significantly outperforming DELLA (0.55/0.38). In particular, AE(Full) achieves a high MAUVE score of 0.69. These results indicate that the prior distribution obtained by adapting to the model's distribution is meaningful and faithful.

The AE models achieve the best results across all of the diversity metrics. Concretely, the AE(EMA) models obtain lower Self-BLEU scores than DELLA by 3.52 and 4.87 on Yelp and Yahoo, respectively. For the Dist score, the performance of the AE(EMA) models surpasses the previous best results by 2.23 and 6.90. The superior diversity demonstrates that the empirical prior is more expressive than the standard Gaussian. In addition, the easier-to-train AE does not affect the capability of the decoder and exhibits better adaptability to the variants of latent variables.

Finally, we investigate the performance of different density estimation methods for the empirical prior. Specifically, AE(Full) and AE(Diag) apply the existing density estimation algorithms[1] with full and diagonal covariance matrixes, respectively. We can see that AE(EMA) achieves comparable performance to its counterpart AE(Diag), indicating that the EMA is an effective approximation of the offline estimation and is more efficient without the additional pass of the training data. Furthermore, the EMA approach offers increased convenience for scaling up to larger models and datasets.

AE(Full) tends to perform better on the quality metrics while performing inferior on diversity. The underlying reason is that the distributions with full covariance matrixes are highly expressive to fit more details, while the diagonal Gaussian is simpler for better generalization. The additional results on SNLI are given in Appendix A.

M	NS	PPL
-	-	6.04*
10	1	8.91
20	1	7.60
50	1	6.84
100	1	6.67
100	5	6.67
100	10	6.67
200	1	6.88

Table 2: Estimated perplexity using the empirical posterior on the Yelp test set. M denotes the number of retrieved latent vectors to estimate the empirical posterior of AE. NS denotes the number of samples to calculate the average PPL.

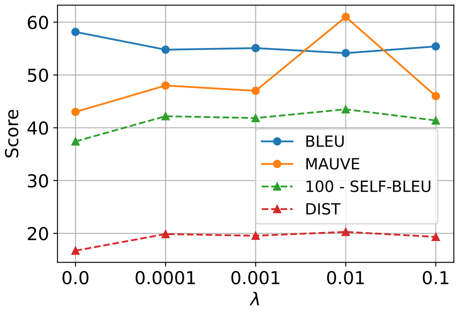


Figure 2: BLEU, MAUVE, Self-BLEU, and DIST on Yelp with different coefficients of the orthogonal regularization loss λ. The higher the scores, the better the performance. We convert all scores into percentages.

5 Analysis

5.1 Empirical Posterior Estimation

The VAE models usually use multiple sampled latent vectors from the posterior distribution to calculate the weighted average perplexity. In Table 2, we investigate the PPL of stochastics sampling from the empirical posterior of the AE model. The results show that the estimated PPL scores are higher than the exact value (6.04) but still lower than the VAE model (e.g., 12.35 achieved by DELLA in Table 1). As M increases from 10 to 100, the estimated PPL decreases. Moreover, VAE typically requires a large NS (e.g., 30) to estimate PPL, while it is not sensitive for the AE to NS.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html

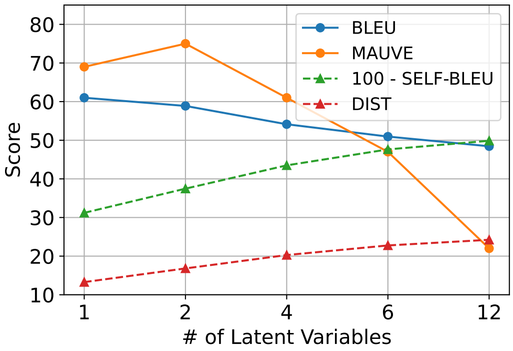


Figure 3: Performance of different numbers of latent vectors. The higher the scores, the better the performance. All scores are displayed in percentages.

Model	Quality		Diversity	
	BLEU↑	MAUVE↑	S-BLEU↓	Dist↑
Uniform	54.14	0.61	56.50	20.28
High	53.69	0.46	55.92	21.02
Low	60.60	0.22	65.93	15.54
Beam(10)	54.14	0.89†	56.50	20.28
Gready	36.10	0.50†	40.00	29.61
TopK(50)	30.36	0.38†	31.29	36.06
TopP(0.95)	30.81	0.40†	31.84	36.24

Table 3: Effect of the positions of latent variables production and decoding strategies on Yelp. S-BLEU denotes the Self-BLEU score. The number with dagger † denotes the MAUVE scores are calculated with a scaling parameter of 3 to help with interpretability.

5.2 Effect of Orthogonal Regularization

In this section, we display the influence of the orthogonal regularization loss. The value of the coefficient λ is chosen from $\{0.0, 0.1, 0.01, 0.001, 0.0001\}$, and we show the generation metrics BLEU and MAUVE, and diversity metrics, (100 - Self-BLEU) and DIST on Yelp in Figure 2. As λ increases to 0.01, the MAUVE score reaches its peak and then decreases when λ increases to 0.1. The orthogonal regularization also improves the Self-BLEU and DIST scores, and the reason can be that the decrease of the overlapping information between latent vectors brought by the regularization leads to a more diverse representation. Notably, we do not fine-tune the coefficient and a value of 0.01 performs consistently well across all of the datasets. This coefficient is also the only hyperparameter we introduce, and it is easier to train the AE model than the VAE models which introduce various training tricks such as KL annealing, BOW loss, and cyclical schedule.

5.3 Number of Latent Variables

AE is easier to train than VAE and may have different requirements for the number of latent variables, which controls the abstraction levels of the input. In this experiment, we investigate its influence on the performance of our AE model. As depicted in Figure 3, increasing the number of latent variables results in lower Self-BLEU scores and higher Dist scores, which means lower similarity between generated sentences. The result is intuitive, and the model can generate a wider range of latent compositions with more latent vectors. However, an excessive number of latent vectors (i.e., 12) can lead to a decrease in generation quality, possibly due to overfitting caused by the relatively small dataset. The choice of the number of latent vectors can influence the trade-off between quality and diversity. Moreover, as mentioned in Section 4.4, the AE model achieves better performance with fewer latent vectors compared to DELLA.

5.4 Position of Latent Variables Production

We employ uniform distribution for the latent variables across the encoder layers and investigate the effect of the position of the latent vector production in this section. The number of latent vectors is set to 4. As shown in the upper part of Table 3, the quality scores of the variant producing latent vectors from the higher layers are inferior to the uniform one especially on the MAUVE score, while the diversity scores are similar. The performance of producing latent vectors from the lower layers is significantly degenerated. The underlying reason is that the semantic information in the shallow encoder layers is not abstract enough to fit with the purpose of learning abstract latent variables.

5.5 Decoding Strategies

The additional latent condition sampled from empirical prior may affect the performance of different decoding strategies. We investigate commonly used decoding strategies consisting of greedy search, beam search, Top-K sampling (Fan et al., 2018), and Top-P sampling(Holtzman et al., 2020). Since the sampling-based decoding algorithms get low MAUVE scores with the default scaling parameter, we decrease it to 3.0 for all of the methods to help with better interpretability following (Pillutla et al., 2021). The results are shown in the bottom half of Table 3. Despite better diversity scores, the sampling-based decoding algorithms archive much lower scores on quality, indicating a tendency to generate sentences different from the

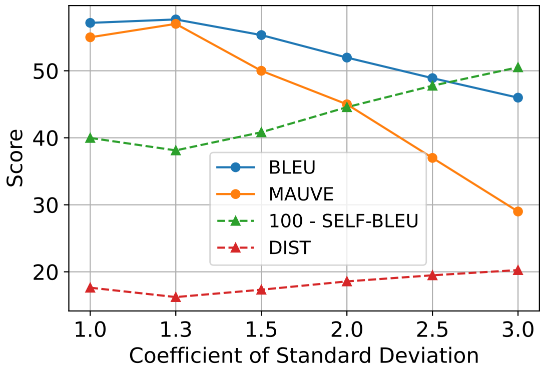
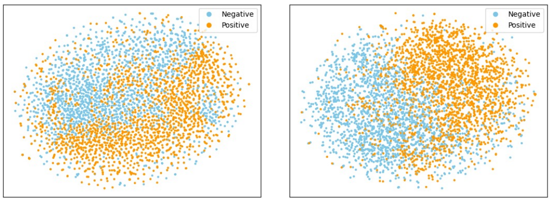


Figure 4: BLEU, MAUVE, and Self-BLEU of generated text using different coefficients of the standard deviation of the DELLA's prior.



(a) DELLA

(b) Ours

Figure 5: T-SNE of latent vectors.

training distribution. The deterministic decoding algorithms are a more suitable choice considering the overall performance, and beam search achieves the best quality-diversity trade-off. Possibly the sampling-based decoding algorithms can perform better when pretraining on massive text with the proposed framework, which is left as future work.

5.6 Stretching Prior of VAE

The high diversity can be attributed to diverse latent variables sampled from the no overly restricted empirical prior. To make DELLA generate more diverse text, we sample latent variables from stretched priors. The result is shown in Figure 4. As we increase the coefficient of the standard deviation, we are more likely to sample latent variables beyond the standard Gaussian prior, and the generated sentences are more diverse, i.e., lower Self-BLEU scores. However, the quality of the generated text becomes worse. This indicates that the decoder of DELLA cannot handle the out-of-distribution latent variables well. By contrast, the scope of exploration in the AE model is more flexible, and the decoder capacity is not affected.

-	Grammaticality	Novelty	Overall
DELLA	17.0%	28.5%	24.5%
AE	35.0%	31.0%	42.5%
No-pref	48.0%	40.5%	33.0%

Table 4: Human evaluation of grammaticality, novelty, and overall quality on SNLI.

6 Human Evaluation

We conduct a human evaluation to compare samples generated by the AE model and the SOTA baseline DELLA using an A/B test (Subramanian et al., 2018). We sample 200 sentences from the generated sentences of each model. We present three annotators with two samples, one from each model, and ask them to indicate their preference based on grammaticality, novelty, and overall quality (Table 4). We can see that the AE model performs better than DELLA. Moreover, the AE model is less prone to generating sentences with grammatical errors, which may be attributed to not affecting the capability of the language model itself.

7 Case Study

Visualization We employ t-SNE to inspect the representation space of DELLA, our AE, and GPT-2. Using the Yelp development set, we categorize 1-star sentences as negative and 5-star sentences as positive. Figure 5 depicts clearer separation in DELLA and AE's latent representations, indicating that the similar latent space to disentangle semantics. By contrast, the sentence representations of GPT-2 obtained by average token representations exhibit entangled semantic discernment (Figure 6 in Appendix), demonstrating the superiority of the conditional language models.

Interpolation Samples of mid-point interpolation between two random sentence pairs are shown in Figure 7 (Appendix F). The interpolated sentences exhibit meaningful semantic interpolation.

8 Conclusion

We investigated training latent conditional language models with autoencoding, which removes the prior and the corresponding restriction in VAE. We found that adapting the prior distribution to the language model can satisfy the need for sampling without affecting the modeling capability of the language model itself. The generated samples from the empirical prior achieve better quality and diversity than the VAE-based models. Future work includes the extension of AE-based language models to large-scale pertaining and instruction fine-tuning, in which the high-level semantic condition of the target sequence may have additional benefits.

Acknowledgements

This work has been supported by the China Strategic Scientific and Technological Innovation Cooperation Project Grant No. 2022YFE0204900 (The Macao counterpart project Grant No. FDCT/0070/2022/AMJ).

9 Limitations

In this study, our focus was primarily on fine-tuning the GPT-2 model to investigate the potential of text AE as a generative model. Pretraining a big text AE model on massive text is an interesting direction, which we leave as future work. Additionally, we identify the exploration of latent conditional language models within the contexts of in-context learning and instruction fine-tuning paradigms as promising avenues for future research.

10 Ethics Consideration

We strictly adhere to the ACL Code of Ethics, ensuring that no private data or non-public information is utilized in this study. To conduct human evaluation, we have engaged three annotators who possess degrees in English Linguistics or Applied Linguistics. We have established a fair compensation rate of $25 per hour for their valuable services.

References

Yoshua Bengio. 2008. Neural net language models. *Scholarpedia*, 3(1):3881.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21. ACL.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. 2016. Importance weighted autoencoders. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2021. Cocon: A self-supervised approach for controlled text generation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Bin Dai and David P. Wipf. 2019. Diagnosing and enhancing VAE models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Shuyang Dai, Zhe Gan, Yu Cheng, Chenyang Tao, Lawrence Carin, and Jingjing Liu. 2021. Apo-vae: Text generation in hyperbolic space. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 416–431. Association for Computational Linguistics.

Xiaoan Ding and Kevin Gimpel. 2021. Flowprior: Learning expressive priors for latent variable sentence models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3242–3258. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Transformer-based conditional variational autoencoder for controllable story generation. *CoRR*, abs/2101.00828.

Xianghong Fang, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Dit-Yan Yeung. 2022. Controlled text generation using dictionary prior in variational autoencoders. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 97–111. Association for Computational Linguistics.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 240–250. Association for Computational Linguistics.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael J. Black, and Bernhard Schölkopf. 2020. From variational to deterministic autoencoders. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Geoffrey Hinton and Ruslan Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jinyi Hu, Xiaoyuan Yi, Wenhao Li, Maosong Sun, and Xing Xie. 2022. Fuse it more deeply! A variational transformer with layer-wise latent variable inference for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 697–716. Association for Computational Linguistics.

Zhiting Hu and Li Erran Li. 2021. A causal lens for controllable text generation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24941–24955.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 424–434. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Diederik P. Kingma, Tim Salimans, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. *ArXiv*, abs/1606.04934.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. 2019. A surprisingly effective fix for deep latent variable modeling of text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3601–3612. Association for Computational Linguistics.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4678–4699. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics.

Andriy Mnih and Karol Gregor. 2014. Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1791–1799. JMLR.org.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Tom Pelsmaeker and Wilker Aziz. 2020. Effective estimation of deep generative language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7220–7236. Association for Computational Linguistics.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. 2021. MAUVE: measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4816–4828.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Rafae Khan, and Jia Xu. 2022. Analyzing encoded concepts in transformer language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3082–3101. Association for Computational Linguistics.

Sandeep Subramanian, Sai Rajeswar, Alessandro Sordoni, Adam Trischler, Aaron C. Courville, and Chris Pal. 2018. Towards text generation with adversarially learned neural outlines. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7562–7574.

Jakub M. Tomczak and Max Welling. 2018. VAE with a vampprior. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR.

Arash Vahdat and Jan Kautz. 2020. NVAE: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. 2019. How does BERT answer questions?: A layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1823–1832. ACM.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315.

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4446–4452. ijcai.org.

Tianming Wang and Xiaojun Wan. 2019. T-CVAE: transformer-based conditioned variational autoencoder for story completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5233–5239. ijcai.org.

Tianyu Yang, Thy Tran, and Iryna Gurevych. 2023. Dior-CVAE: Pre-trained language models and diffusion priors for variational dialog generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4718–4735, Singapore. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3881–3890. PMLR.

Meng-Hsuan Yu, Juntao Li, Danyang Liu, Bo Tang, Haisong Zhang, Dongyan Zhao, and Rui Yan. 2020. Draft and edit: Automatic storytelling through multi-pass hierarchical conditional variational autoencoder. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020,* *New York, NY, USA, February 7-12, 2020*, pages 1741–1748. AAAI Press.

Yizhe Zhang, Guoyin Wang, Chunyuan Li, Zhe Gan, Chris Brockett, and Bill Dolan. 2020. POINTER: constrained progressive text generation via insertion-based generative pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8649–8670. Association for Computational Linguistics.

Qile Zhu, Wei Bi, Xiaojiang Liu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. 2020. A batch normalized inference network keeps the KL vanishing away. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2636–2649. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 1097–1100. ACM.

A Additional Experiment Results

The results on SNLI are shown in Table 5. The AE models also achieve better performance than the VAE baselines.

B Computational Details

We implemented our model and conducted experiments using the Huggingface Transformers library version 4.29.0. The experiments were performed on NVIDIA A100 GPU. All experimental results are trained and tested in a single run with seed 42 following Hu et al. (2022). For parameter sizes, Optimus uses BERT as the encoder, while the others use GPT-2. We also follow DELLA to share the parameters of the encoder and decoder. The parameter sizes are listed as follows: GPT-2 (124M), Optimus (234M), DELLA (193M), and AE (125M).

C Dataset Licenses

For the licenses of the datasets, Yelp uses its own license, Yelp Data Agreement, which allows their data for academic use. SNLI uses CC BY-SA 4.0. The license for the Yahoo Dataset is not found.

D Metrics

Here, we provide comprehensive details about the evaluation metrics.

Activate Units (AU) (Burda et al., 2016) refers to the number of unique units (neurons or dimensions) in the latent space that are activated during the encoding of input data. Monitoring activated units can provide insights into the diversity and richness of the learned latent representations. Higher activation of different units suggests a more expressive latent space.

BLEU (Papineni et al., 2002) measures the n-gram overlap between generated sequences and reference sequences. In the context of text VAE, all samples in the test set are considered references for each generated example.

MAUVE (Pillutla et al., 2021) directly evaluates the learned distribution from a text generation model against the distribution of human-written text using divergence frontiers, which has been demonstrated to have a higher correlation with human judgments. It relies on pre-trained language models like GPT-2.

Self-BLEU (Zhu et al., 2018) is a variation of BLEU where the metric is computed by comparing a set of generated samples against each other rather than reference texts. This metric helps to evaluate the diversity of generated outputs. Lower Self-BLEU scores indicate more diverse and varied generated text.

Dist (Li et al., 2016) measures the proportion of distinct n-grams in generated samples. A higher proportion of distinct n-grams indicates more diverse samples. We use bigrams following previous studies.

Jaccard Similarity (JS) (Wang and Wan, 2018) measures the similarity between two sets by calculating the intersection divided by the union of the sets.

E Visualization of Sentence Representations

The t-SNE plot depicting the GPT-2 sentence representations is presented in Figure 6. It can be observed that the representation space is more entangled compared to AE and VAE models.

F Case Study

VAE has been shown good at linear interpolating between latent vectors. We also conduct the interpolation with AE on SNLI. The examples in Figure

Model	**PPL↓**	**AU↑**	**Quality**		**Diversity**	
			BLEU↑	**MAUVE↑**	**Self-BLEU↓**	**Dist↑**
SNLI						
GPT-2(small)						
GPT-2	20.19	-	**63.57**	0.71	75.34	19.11
Optimus	16.67	-	-	-	-	-
Embedding	13.79	20	59.26	0.72	65.59	20.89
Memory	13.78	10	62.80	0.67	54.59	21.87
Softmax	14.21	16	60.51	0.71	71.84	18.59
DELLA	5.13	23	62.94	0.69	36.85	32.61
AE(EMA)	**1.76**	**32**	53.73	0.70	**36.06**	**34.95**
AE(Full)			61.36	0.75	41.50	30.67
AE(Diag)			60.26	0.66	40.71	30.50
GPT-2(medium)						
AE(EMA)	2.18	**32**	58.19	0.82	40.75	31.34
AE(Full)			62.16	**0.83**	42.68	29.34
AE(Diag)			62.28	0.75	42.93	28.61

Table 5: Evaluation results on SNLI. The best results are highlighted in bold.

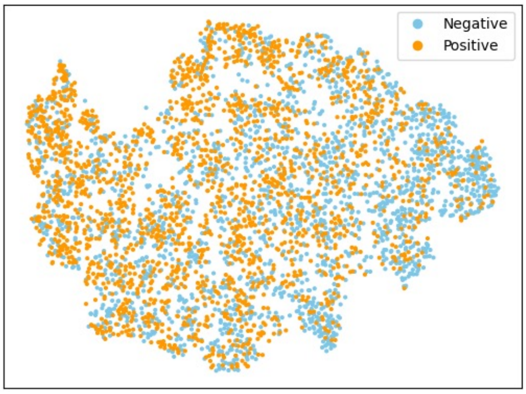


Figure 6: T-SNE plot for GPT-2.

S1: the man works on the net .
S1-*-S2: the man is working on the tube .
S2: the man is waiting at the tube shop .

S1: children are looking to see if the coast is clear .
S1-*-S2: a man is looking to see a bike pass through the fair .
S2: a man is riding a bike through a park .

S1: a man is taking beautiful photos by the river .
S1-*-S2: a man is enjoying the beautiful sun by the river .
S2: a goat is enjoying the sun on the farm .

Figure 7: Case study.

7 show that the interpolated sentences exhibit meaningful semantic interpolation.