

Finetuning Pretrained Transformers into Variational Autoencoders

Seongmin Park Jihwa Lee

ActionPower

Seoul, Republic of Korea

{seongmin.park, jihwa.lee}@actionpower.kr

Abstract

Text variational autoencoders (VAEs) are notorious for *posterior collapse*, a phenomenon where the model’s decoder learns to ignore signals from the encoder. Because posterior collapse is known to be exacerbated by expressive decoders, Transformers have seen limited adoption as components of text VAEs. Existing studies that incorporate Transformers into text VAEs (Li et al., 2020; Fang et al., 2021) mitigate posterior collapse using massive pretraining, a technique unavailable to most of the research community without extensive computing resources. We present a simple two-phase training scheme to convert a sequence-to-sequence Transformer into a VAE with just *finetuning*. The resulting language model is competitive with massively pretrained Transformer-based VAEs in some internal metrics while falling short on others. To facilitate training we comprehensively explore the impact of common posterior collapse alleviation techniques in the literature. We release our code for reproducibility¹.

1 Introduction

Properly tamed latent models offer explainable and interpolatable representations of observed data. Recent works have shown such models to be especially useful in unsupervised learning settings. He et al. (2019) adapt a generative latent text model for successful unsupervised text style transfer and machine translation. Li et al. (2020) achieve superior language modeling performance against common conditional counterparts.

A popular variant of deep latent models is the variational autoencoder (VAE) (Kingma and Welling, 2014). For each observed \mathbf{x} , the model assumes the existence of a corresponding multi-dimensional latent vector \mathbf{z} . Since the log evidence

$\log p(\mathbf{x})$ is intractable for most interesting problems, the training process for VAEs opts instead to minimize the log evidence lower bound (ELBO):

$$E_{z \sim q(z|x)}[\log(p(\mathbf{x}|z))] - D_{KL}(q(z|x)||p(z)) \quad (1)$$

$q(z|x)$ is a tractable, assumed posterior commonly modeled with a parametrized encoder $q_\phi(z|x)$, while $p(\mathbf{x}|z)$ is the likelihood parametrized with a decoder $p_\theta(\mathbf{x}|z)$ that optimizes against reconstruction loss. While effective in theory, a common empirical challenge VAEs present during training is posterior collapse – a phenomenon where the decoder ignores the latent signal from \mathbf{z} (and thus the originating input) during reconstruction. Posterior collapse can be diagnosed by checking if $D_{KL}(q(z|x)||p(z))$ tends to zero during training.

After Bowman et al. (2016) adopted VAE for text, subsequent studies have been introduced with attempts to mitigate posterior collapse in VAE language models (LMs). However, the brittle training process of VAE LMs remains an unsolved problem.

Li et al. (2020) present a method to utilize deep Transformer (Vaswani et al., 2017) models as components of VAE LMs. Transformer-based VAEs tap into the state-of-the-art capabilities of Transformers while retaining representational advantages of VAE LMs. The paper mitigates posterior collapse by massive pretraining and a cyclical annealing schedule (Fu et al., 2019).

While the study presents a promising outlook for Transformer VAEs, the suggested method is not accessible to researchers who lack access to large, target-domain-specific corpora or the computing power for massive LM pretraining. Therefore, a demand arises for a way to finetune an existing Transformer model into a VAE LM with limited resources. Our research attempts to fill this gap in the literature, and makes the following contributions:

- We present a simple but reliable (as replicated across several datasets) scheme to teach latent

¹<https://github.com/seongminp/transformers-into-vaes>

structure to a pretrained Transformer model by just finetuning.

- We convert a pretrained sequence-to-sequence Transformer into a VAE, instead of using two separate encoder-only (Devlin et al., 2019) and decoder-only (Radford et al., 2019) Transformers as in previous literature. This eliminates the need to maintain separate tokenizers and configurations for encoder and decoder.
- We conduct ablation studies and extensive experiments to gauge the effectiveness of commonly used posterior collapse mitigation methods in taming Transformer VAEs.

The resulting model extends existing Transformer architectures and can be initialized from pretrained non-latent model checkpoints.

2 Background

2.1 Transformer Text VAEs

Most VAE LMs employ recurrent neural networks (RNNs) as encoders and decoders. This is in part because enforcing a latent bottleneck layer undermines the effectiveness of encoder-decoder cross-attention in Transformers, and in significant part due to the co-occurrence of posterior collapse and powerful and deep decoder layers. Li et al. (2020) overcome such training difficulties by massively increasing the number of training samples (104,213,036 sentences) for LM pretraining.

Liu and Liu (2019) and Fang et al. (2021) also find success with Transformer VAEs for text generation. To avoid posterior collapse, Fang et al. (2021) follow the exact cyclic KL mitigation approach as that of Li et al. (2020), while Liu and Liu (2019) introduce noise to network input.

2.2 Techniques to mitigate posterior collapse

This study identifies and explores the effect of popular posterior collapse mitigation methods in low-resource Transformer VAE training. We do not examine importance-weighted autoencoders (Burda et al., 2016) and semi-amortized autoencoders (Kim et al., 2018) to limit the scope of our experiments to unsophisticated prior distributions.

2.2.1 KL Weighting / Annealing

Bowman et al. (2016) increases the KL term of the ELBO from zero to its full value during early stages of training, where the decoder learns to simply treat

latent signal \mathbf{z} as noise. Fu et al. (2019) extend this technique by cyclically manipulating the weight of the KL term. β -VAE (Higgins et al., 2017) and Yan et al. (2020) adopt a similar approach.

2.2.2 Encoder warm-up (Li et al., 2019)

We train the network without the KL term of the ELBO and retain encoder weights before jointly training the whole network.

2.2.3 Input text denoising (Shen et al., 2020)

Denoising text inputs by deleting random tokens motivate autoencoders (AEs) to learn better latent representations. Our study compares 0%, 15%, and 40% deletion noising schemes.

2.2.4 KL thresholding (Kingma et al., 2016)

KL-thresholding enforces a minimum λ for each dimension of the KL term in the ELBO:

$$\mathcal{L}_{DKL} = \sum_i \max[\lambda, D_{kl}(q_\phi(z_i|x)||p(z_i))] \quad (2)$$

where z_i is a single dimension of \mathbf{z} .

2.2.5 Encoder pooling (Long et al., 2019)

Instead of using the last hidden state as encoder output, averaging or taking the maximum of all encoder hidden states results in a more diverse latent representation. We experiment with both mean- and max-pooling schemes from the encoder.

3 Model architecture

We extend the T5 architecture (Raffel et al., 2020) into a VAE. We modify a popular pretrained T5 model (Wolf et al., 2020) that deviates minimally from the original Transformer (Figure 1).

Hidden states from all layers of T5’s encoder $q_\phi(z|x)$ are mean- or max-pooled into a vector $h_{pooled} \in R^H$, where H is the encoder’s hidden dimension.

Assumed prior $q(z)$ ’s mean μ and log variance σ vectors of dimension L is obtained from h_{pooled} :

$$\mu = h_{pooled}W_\mu, \log\sigma = h_{pooled}W_\sigma \quad (3)$$

where $W_\mu, W_\sigma \in R^L$.

As in a standard VAE, a stochastic latent vector \mathbf{z} is sampled using the reparameterization trick to enable back-propagation through sampling:

$$\mathbf{z} = \mu + \sigma \odot \epsilon, \epsilon \sim \mathcal{N}(0, 1) \quad (4)$$

We pass \mathbf{z} into the decoder $p_\theta(x|\mathbf{z})$ as the only component of decoder’s cross attention. Our

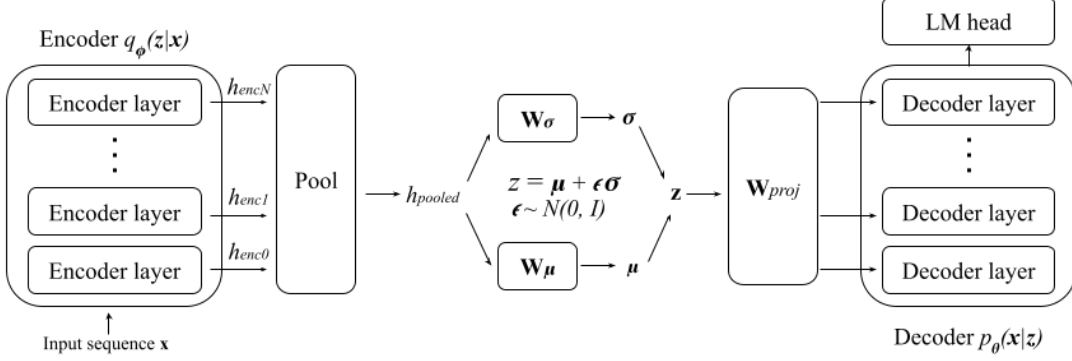


Figure 1: Transformer VAE architecture. A "bottleneck" step (W_σ and W_μ) is placed between the encoder and the decoder of T5. Latent information from pooled encoder hidden states is captured in the bottleneck layer before being passed to the decoder. The network is optimized against regularization loss in the bottleneck and reconstruction loss at the decoder.

method injects \mathbf{z} into every layer of the decoder as in previous literature (Li et al., 2020; Fang et al., 2021), but deviates in two important ways: first, we pass \mathbf{z} as the sole key and value of encoder-decoder cross attention, instead of self-attention; second, we project \mathbf{z} into the correct dimension ($L \times A \times S$, where L is the decoder layer count, A is the number of attention heads, and S is the embedding dimension per head) with a feed-forward network, instead of taking a copy of \mathbf{z} to inject to each decoder layer.

$$(K_{ca}, V_{ca}) = (\mathbf{z}W_{proj}, \mathbf{z}W_{proj}) \quad (5)$$

where $W_{proj} \in \mathbb{R}^{L \times A \times S}$ and K_{ca} and V_{ca} are key and value in decoder cross-attention.

4 Experiments

During preliminary experiments, posterior collapse was observed in all training schemes without encoder warmup training. The decoder learns to ignore the initially noisy input signal from the encoder. Thus, we compose our finetuning method in two separate phases.

Phase 1 - Encoder warmup: Weight of KL loss is set to zero, making our model’s objective function similar to that of an AE. Different input denoising percentages, encoder pooling strategies, latent dimension sizes, and decoder freezing configurations are compared.

Phase 2 - Full finetuning: KL loss is reinstated and full VAE training are conducted. We compare different input denoising percentages, encoder pooling strategies, KL annealing schedules, and KL thresholds.

We run our proposed two-phase finetuning training scheme on four standard VAE LM benchmark

datasets: PTB (Marcus et al., 1993), SNLI (Bowman et al., 2016), Yahoo (Yang et al., 2017), and Yelp (Shen et al., 2017).

Following Li et al. (2020) and Li et al. (2019), we perform intrinsic evaluation of our proposed Transformer VAE architecture. We report perplexity (PPL), KL-divergence between model posterior and assumed posterior (KL), and negative ELBO on the test set. To assess the quality of learned latent codes, we also report mutual information (MI) (Hoffman and Johnson, 2016) and the number of active units (AU) (Burda et al., 2016). MI measures the dependence of latent codes to encoder input. AU measures the covariance between encoder input and latent codes.

Experimental hyperparameters such as specific annealing schedules and training epochs per phase are detailed in the appendix.

5 Results

5.1 Phase 1

We find that freezing the decoder and the memory projection layer W_{proj} while training with an AE objective is crucial in learning meaningful encoder outputs. Denoising is important for datasets with longer inputs (Yahoo, Yelp), but not critical in datasets with shorter input lengths (PTB, SNLI). Mean-pooling encoder hidden states presents a trade-off between MI and AU. Max-pooling consistently learns more informative encoder representations. Changes in MI and AU during training is illustrated in Figure 2.

Latent dimensions of 64 and 128 were also tested. Increasing the latent dimension did not necessarily boost representational quality in terms

Model	PPL↓	KL	-ELBO↓	MI↑	AU↑
Optimus ($\lambda = 0.5$) (Li et al., 2020)	23.11	17.45	301.21	8.85	32
GPT-2 (Radford et al., 2019)	22.00	-	-	-	-
Encoder pretraining ($\lambda = 3$) (Li et al., 2019)	59.24	7.44	328.73	6.41	32
Ours (Max pool)	20.90	0.21	343.02	0.04	0
Ours (Max pool + Denoise)	30.13	41.49	301.86	1.32	24
Ours (Max pool + Denoise + KLT)	60.44	119.89	223.69	4.73	29
Ours (Max pool + Denoise + KLT + Deep)	54.40	155.50	140.57	5.43	28

Table 1: Phase 2 results on Yahoo. Due to space constraints, we report experimental results on other datasets in the appendix. Results on baselines are quoted from Li et al. (2020) and (Li et al., 2019). KLT denotes KL thresholding with $\lambda = 3$. Our models are finetuned from a pretrained 6-layer T5, except the deep variant with 12 layers.

of AU percentage. For latent dimensions of 32 and 64, 90% of latent dimension units were activated in best-performing models. For latent dimension of 128, around 60% of latent units were active.

Another interesting observation is that KL divergence on the validation set, although not part of the AE training objective, plateaus after repeated training. We regard this phenomenon as the signal of convergence in terms of representation quality.

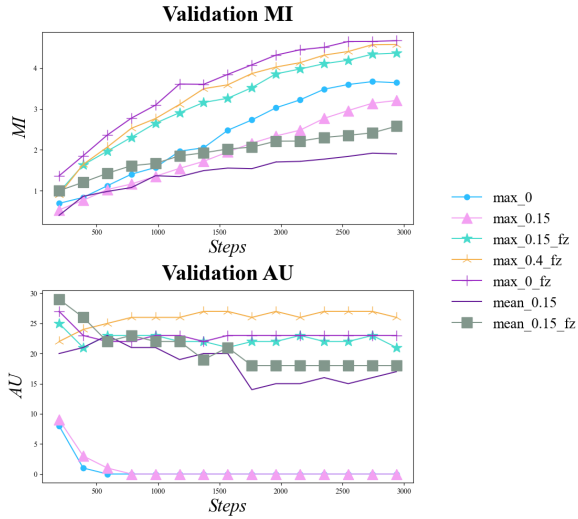


Figure 2: Phase 1 training on Yahoo. Labels are in the form {pooling strategy}_{denoise percentage}_{decoder frozen}.

5.2 Phase 2

We observe, as in previous literature, a trade-off between language modeling PPL and representation quality metrics (MI and AU). This trade-off is exacerbated when using KL thresholding. While KL thresholding does significantly increase latent representation capabilities, it is not in itself sufficient in preventing posterior collapse.

Denoising and encoder pooling configurations

display the same characteristics as in Phase 1. No version of the experiment existed where cyclical annealing schedule was able to prevent posterior collapse, a result not in accordance with Li et al. (2020). Figure 3 illustrates the training progression of Phase 2.

We also experimented with increasing model depth from 6 layers to 12 layers. Our proposed two-phase training scheme prevents posterior collapse for deeper models as well, resulting in higher performance in most metrics compared to 6-layer models. Results are reported in Table 1. Note that lower PPL does not necessarily indicate better language modeling capabilities, since models with collapsed posterior display better PPL.

Rows with KL above zero indicate successful aversion of posterior collapse.

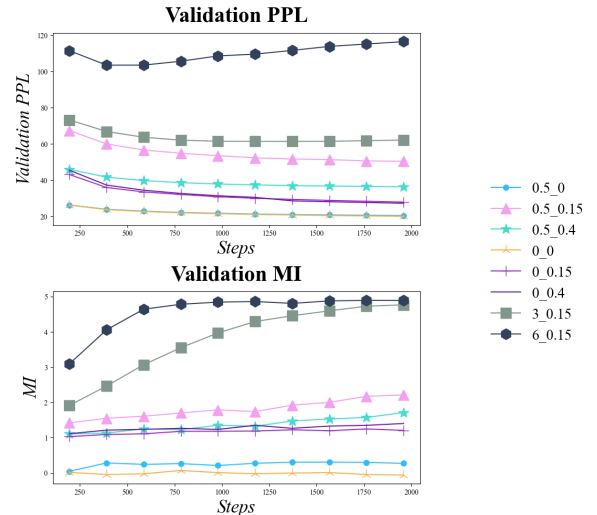


Figure 3: Phase 2 training on Yahoo. Labels are in the form {KL threshold}_{denoise percentage}. Encoder hidden states in plotted experiments were max-pooled.

In the literature, no consensus yet exists on the optimal value of KL in training VAEs. Overall, we

find that a denoising scheme between 0.15 and 0.4 in both phases, coupled with a low (0.5) KL threshold strikes a good balance between reconstruction and latent representation quality.

6 Conclusions and Future Work

This paper explores common methods in the literature for combatting posterior collapse, and the extent to which they help in teaching latent information to pretrained Transformer models.

Comprehensive experiments show that commonly employed posterior collapse mitigation techniques provide meaningful benefits in transforming existing language models into latent-aware architectures. Among the tested procedures, we find that Li et al. (2019)’s two-step training, coupled with Shen et al. (2020)’s denoising through token deletion, was the most impactful in mitigating posterior collapse. However, language models obtained via only finetuning exhibit consistent trade-offs between their latent representation metrics (MI, AU) and language model metrics (PPL). Optimizing our model to be competitive with massively pretrained baselines in one of the two metrics results in the model falling behind in the other.

We also find that increasing training epochs further improves the impact of tested techniques, a result consistent with previous literature on large-scale text VAE pretraining.

From our experiments, we identify several questions to be answered by future research. The impact of homogenizing finetuning (as suggested in this paper) and original pretraining objectives on language model metrics has to be further explored. While the original T5 architecture was also pretrained with a self-supervised denoising scheme, the model employs mask tokens for denoising, contrary to simple token deletions suggested by this paper.

Our findings also highlight the need for an established heuristic to interpret the quality of latent representations learned by language models. The research community has yet to decide on the optimal value of KL-divergence between the assumed prior and the model posterior to target during text VAE training. Empirical guidelines to dictate even a vague threshold for the KL-divergence, below which we declare the occurrence of posterior collapse, will help both training and evaluation of latent-aware language models.

References

- Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.
- Yuri Burda, Roger B Grosse, and Ruslan Salakhutdinov. 2016. Importance weighted autoencoders. In *ICLR (Poster)*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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. *arXiv preprint arXiv:2101.00828*.
- 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, Volume 1 (Long and Short Papers)*, pages 240–250.
- Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.
- Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew 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.
- Matthew D Hoffman and Matthew J Johnson. 2016. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1.
- Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. 2018. [Semi-amortized variational autoencoders](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2678–2687. PMLR.

- 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*.
- Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. [Improved variational inference with inverse autoregressive flow](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- 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)*, pages 3603–3614, Hong Kong, China. Association for Computational Linguistics.
- Chunyu Li, Xiang Gao, Yuan Li, Baolin Peng, Xiu-jun 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)*, pages 4678–4699.
- Danyang Liu and Gongshen Liu. 2019. A transformer-based variational autoencoder for sentence generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.
- Teng Long, Yanshuai Cao, and Jackie Chi Kit Cheung. 2019. Preventing posterior collapse in sequence vaes with pooling. *arXiv e-prints*, pages arXiv–1911.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. [Building a large annotated corpus of English: The Penn Treebank](#). *Computational Linguistics*, 19(2):313–330.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6833–6844, Red Hook, NY, USA. Curran Associates Inc.
- Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi Jaakkola. 2020. Educating text autoencoders: Latent representation guidance via denoising. In *International Conference on Machine Learning*, pages 8719–8729. PMLR.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Trans-formers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Chaochao Yan, Sheng Wang, Jinyu Yang, Tingyang Xu, and Junzhou Huang. 2020. [Re-balancing variational autoencoder loss for molecule sequence generation](#). In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, BCB ’20*, New York, NY, USA. Association for Computing Machinery.
- 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 - Volume 70, ICML’17*, page 3881–3890. JMLR.org.

A Phase 2 results on PTB, Yelp, and SNLI

Model	PPL↓	KL	-ELBO↓	MI↑	AU↑
Optimus ($\lambda = 0.5$) (Li et al., 2020)	26.69	15.72	96.82	7.64	32
GPT-2 (Radford et al., 2019)	24.23	-	-	-	-
Encoder pretraining ($\lambda = 3$) (Li et al., 2019)	96.75	3.85	101.56	3.19	32
Ours (Max pool)	51.60	0.09	104.14	0	0
Ours (Max pool + Denoise)	57.69	3.05	101.17	11	
Ours (Max pool + Denoise + KLT)	250.44	41.93	62.25	2.17	24
Ours (Max pool + Denoise + KLT + Deep)	705.73	84.28	4.95	13	

Table 2: Phase 2 results on PTB

Model	PPL↓	KL	-ELBO↓	MI↑	AU↑
Optimus ($\lambda = 0.5$) (Li et al., 2020)	22.79	15.09	344.10	9.13	32
GPT-2 (Radford et al., 2019)	23.40	-	-	-	-
Encoder pretraining ($\lambda = 3$) (Li et al., 2019)	-	-	-	-	-
Ours (Max pool)	21.65	0.25	404.54	0	0
Ours (Max pool + Denoise)	39.09	77.85	327.17	1.06	26
Ours (Max pool + Denoise + KLT)	86.71	182.24	223.34	5.46	27
Ours (Max pool + Denoise + KLT + Deep)	53.05	178.48	166.15	5.55	10

Table 3: Phase 2 results on Yelp.

Model	PPL↓	KL	-ELBO↓	MI↑	AU↑
Optimus ($\lambda = 0.5$) (Li et al., 2020)	16.67	16.35	38.50	8.89	32
GPT-2 (Radford et al., 2019)	20.24	-	-	-	-
Encoder pretraining ($\lambda = 3$) (Li et al., 2019)	21.23	5.86	33.87	5.25	32
Ours (Max pool)	12.79	0.11	34.61	0.06	0
Ours (Max pool + Denoise)	15.16	2.41	32.32	0.194	7
Ours (Max pool + Denoise + KLT)	85.88	26.12	8.62	0.88	13
Ours (Max pool + Denoise + KLT + Deep)	2358.31	74.95	43.77	5.27	17

Table 4: Phase 2 results on SNLI.

B Experimental details

For all experiments we used a AdamW optimizer (Loshchilov and Hutter, 2019) with a starting learning rate of 1×10^{-3} , $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-3}$. The linear KL annealing schedule we used was as follows:

$$KL\ weight = \frac{current\ global\ step}{steps\ per\ epoch * 50} \quad (6)$$

Our slower, linear KL annealing schedule of 0 to 1 over 50 epochs yielded better empirical results than the linear schedule used in Li et al. (2019) (0 to 1 over 10 epochs). We attribute this result to the

small number of training samples in our experiments.

We train for 5 epochs on Phase 1, and 3 epochs on Phase 2. While further training leads to increased MI and AU, we limit the number of epochs to conform to the spirit of this study, which is to learn latent representations with minimal training. The 5 epoch limit on Phase 1 was empirically determined as the point where encoder MI begins to plateau. Most experiments were conducted with z dimension of 32 for comparison with previous literature.