VOLTA: Improving Generative Diversity by Variational Mutual Information Maximizing Autoencoder

Yueen Ma[1], **Dafeng Chi**[2], **Jingjing Li**[1], **Kai Song**, **Yuzheng Zhuang**[2], **Irwin King**[1]
The Chinese University of Hong Kong[1]
Huawei Noah's Ark Lab[2]
{yema21, lijj, king}@cse.cuhk.edu.hk
{chidafeng1, zhuangyuzheng}@huawei.com
songkai.neu@gmail.com

Abstract

The natural language generation domain has witnessed great success thanks to Transformer models. Although they have achieved state-of-the-art generative quality, they often neglect generative diversity. Prior attempts to tackle this issue suffer from either low model capacity or over-complicated architectures. Some recent methods employ the VAE framework to enhance diversity, but their latent variables fully depend on the input context, restricting exploration of the latent space. In this paper, we introduce VOLTA, a framework that elevates generative diversity by bridging Transformer 1 with VAE via a more effective cross-attention-based connection, departing from conventional embedding concatenation or summation. Additionally, we propose integrating InfoGAN-style latent codes to enable input-independent variability, further diversifying the generation. 2 Moreover, our framework accommodates discrete inputs alongside its existing support for continuous inputs. We perform comprehensive experiments with two types of Transformers on six datasets from three different NLG tasks to show that our approach can significantly improve generative diversity while maintaining generative quality.

Context	Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto ······
Q	What type of statue is on the main building?
A	golden statue of the Virgin Mary
Q	What is the name of the copper statue on the main building?
A	a copper statue of Christ with arms upraised with ···
Q	What is next to the main building?
A	Grotto

Table 1: Examples of generation diversity by VOLTA on the QAG task. Our framework enables generating three distinct question-answer pairs.

1 Introduction

The rapid advancement of Natural Language Generation (NLG) has been propelled by the remarkable success of Transformer models, including the notable series of GPT models (Radford et al., 2018, 2019; Brown et al., 2020; Ouyang et al., 2022; OpenAI, 2023b,a), T5 (Raffel et al., 2020), OPT (Zhang et al., 2022), and LLaMA (Touvron et al., 2023), etc. While they have demonstrated unparalleled proficiency in autoregressive text generation (Li et al., 2020b; Hu et al., 2022b; Li et al., 2022; Qiu et al., 2024), they predominantly focus on learning to reassemble text from large corpora with high generative quality. However, the pursuit of generative diversity remains a critical yet underexplored frontier in NLG. Generative diversity is distinct from mere paraphrasing, as it encompasses not only altered syntax but also varied semantics. Early attempts, such as diverse beam search (Vijayakumar et al., 2018), have made strides in enhancing diversity by modifying the decoding process. Nonetheless, these methods often fall short in enhancing the model itself, limiting their ability to significantly improve diversity.

They just focus on quality of generation

Variational Autoencoder (VAE) (Kingma and Welling, 2014) offers a framework addressing the low-diversity issue. By encoding inputs into lower-dimensional latent variables, VAE introduces the opportunity to diversify the decoding process: perturbing these latent variables allows generated sentences to deviate from annotated ones, thereby enhancing diversity. However, prior attempts like Info-HCVAE (Lee et al., 2020), utilizing LSTM-based VAEs, inherit limitations associated with LSTMs. While Transformers have emerged as the mainstream network, integrating them into the VAE framework poses challenges due to the parallelized self-attention mechanism. More pre-

LSTM limitation

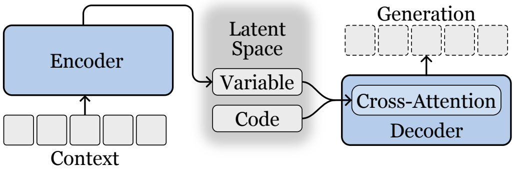

Figure 1: The overview of VOLTA. The encoder encodes the context into VAE latent variables. The variables, augmented with InfoGAN-style **latent codes**, can be continuous or discrete based on the input type. Subsequently, they are connected to the decoder through the **cross-attention** mechanism. Leveraging the variability inherent in the latent space, the decoder generates diverse content conditioned on the context.

VAE
en → Bert
de → GPT2

cisely, this complexity arises from inserting a bottleneck layer of latent variables between Transformer layers, as the embeddings of the entire sequence pass through the model simultaneously. Optimus (Li et al., 2020a), pioneering the fusion of Transformers with VAEs, adopts BERT (Devlin et al., 2019) as the VAE encoder and GPT-2 (Radford et al., 2019) as the VAE decoder. Subsequent works attempt to improve upon Optimus (Hu et al., 2022a; Tu et al., 2022; Deng et al., 2023), yet they fall short in addressing its three major drawbacks. Firstly, it introduces embedding concatenation and summation to connect latent variables to the decoder, with Optimus performing optimally only upon their combined use. In contrast, our novel cross-attention-based connection proves more effective. Secondly, Optimus' model architecture is overly intricate. It relies on two distinct Transformer models, necessitating two unique tokenizers and extensive pretraining. We streamline this complexity by either employing a shared Transformer decoder as the backbone network or leveraging an encoder-decoder Transformer model. This renders our framework compatible even with Large Language Models (LLMs) such as LLaMA (Touvron et al., 2023) or GPT-4 (OpenAI, 2023a). Lastly, while Optimus solely handles continuous latent variables, VOLTA expands its scope to cover discrete inputs by encoding them into discrete latent variables, enriching the model's generalizability.

?

- points of optimus

The VAE framework offers increased generative diversity, yet its input-dependent latent variables limit exploration within the latent space, restricting the model's ability to generate a wider array of diverse content. In pursuit of an input-independent approach to vary the generation process, we propose attaching latent codes to VAE latent variables, inspired by InfoGAN (Chen et al., 2016). Our method employs the Variational Mutual Information Maximization (VMIM) objective to encourage the decoder to autonomously identify distinct semantic features via latent codes. Consequently, this enables more variability in generated content without any reliance on the input. To the best of our knowledge, our work represents the first utilization of latent codes within NLG.

sth like perturbation

transformed representation of input

Our framework, dubbed **VOLTA** (**V**ariati**O**nal Mutua**L** Informa**T**ion Maximizing **A**utoencoder), derives its name from its adherence to the Variational Autoencoder framework and the incorporation of the Variational Mutual Information Maximization objective from InfoGAN. To validate the effectiveness of VOLTA, we benchmark it against state-of-the-art baseline models across six datasets from three representative NLG tasks: language modeling, question-answer generation, and dialog response generation. We also conduct comprehensive ablation studies to examine the impact of the different components of VOLTA.

The main contributions of this paper are:

- VOLTA proposes a novel cross-attention mechanism to integrate Transformer with VAE. It exhibits generalizability to both continuous or discrete latent variables and various Transformer architectures, including decoder-only or encoder-decoder Transformers.
- To attain input-independent variability, we propose attaching InfoGAN-style latent codes to VAE latent variables.
- Comprehensive experimental results on six datasets spanning three distinct NLG tasks validate the efficacy of our model in enhancing generative diversity while upholding quality.

2 Related Work

In recent years, a multitude of Transformer-based models has emerged, such as the GPT series (Radford et al., 2018, 2019; Brown et al., 2020; Ouyang et al., 2022; OpenAI, 2023b,a), T5 (Raffel et al., 2020), OPT (Zhang et al., 2022), and LLaMA (Touvron et al., 2023), etc. These models are primarily trained to optimize the alignment between generated content and annotations, often prioritizing quality over diversity in the generative process.

Variational Autoencoders (VAEs) (Kingma and Welling, 2014) represent a powerful approach to diverse generation in NLG. They diverge from Autoencoders (Hinton and Salakhutdinov, 2006) by introducing low-dimensional latent variables. Originally applied in computer vision, VAEs were later adapted for natural language processing. Early attempts, such as Info-HCVAE (Lee et al., 2020), employed LSTMs (Hochreiter and Schmidhuber, 1997) as both encoder and decoder, achieving diversity in question-answer generation (QAG). However, these LSTM-based models suffered from architectural complexities, utilizing separate LSTM modules for encoding and decoding context, questions, and answers. Optimus (Li et al., 2020a) addressed some of these challenges by using BERT (Devlin et al., 2019) as the encoder and GPT-2 (Radford et al., 2019) as the decoder, surpassing LSTM-based models in VAE language modeling. Subsequent models like VarMAE (Hu et al., 2022a) focused on applying VAEs in language understanding, while RegaVAE (Deng et al., 2023) attempted augmentation through retrieval methods, and AdaVAE (Tu et al., 2022) explored the usage of two adaptive GPT-2 models. Our VOLTA model further simplifies the architecture by leveraging a shared backbone network or utilizing an encoder-decoder Transformer model.

In pursuit of more variability and subsequently, higher diversity, several methods have employed unique strategies such as special prompt tokens or control phrases. These include SimpleTOD (Hosseini-Asl et al., 2020), CTRL (Keskar et al., 2019), Soloist (Peng et al., 2021), CGRG (Wu et al., 2021), and MEGATRON-CNTRL (Xu et al., 2020). Dathathri et al. (2020) proposed the Plug and Play Language Model, which guides language generation by plugging simple attribute classifiers into existing language models. InfoGAN (Chen et al., 2016) originally controlled image generation using latent codes trained with the Variational Mutual Information Maximization (VMIM) objective. In computer vision, attempts to merge InfoGAN with VAE for controllable generative models have resulted in models like VAE-Info-cGAN (Xiao et al., 2020) and InfoVAEGAN (Ye and Bors, 2021). However, InfoVAE (Zhao et al., 2019), InfoMax-VAE (Lotfi-Rezaabad and Vishwanath, 2020), Melis et al. (2022), and VAE-MINE (Qian and Cheung, 2019) applied VMIM to VAE to address the latent variable collapse problem rather than focusing on improving variability. To the best of our knowledge, our model is the first to integrate Transformer models with the VAE and InfoGAN frameworks in Natural Language Generation (NLG). Although we focus on diversity in this paper, other aspects of NLG are also worth exploring in the future (Song et al., 2023c,a,b; Ma et al., 2023), such as multi-modality, bias, and fairness.

check

3 Our Method

Our VOLTA framework is meticulously designed to facilitate diverse generation, leveraging latent variables from the VAE framework (Kingma and Welling, 2014) in conjunction with InfoGAN-style latent codes (Chen et al., 2016). Initially, VOLTA encodes the input into latent variables. Subsequently, by sampling new latent variables, slight alterations in the decoded content can be achieved, promoting greater diversity. Differing from VAE latent variables, InfoGAN-style latent codes operate independently of input, providing the freedom to explore a broader latent space. This distinct attribute offers an alternative avenue to introduce increased variability within the generated sequences. Figure 1 includes an overview of VOLTA.

3.1 Preliminaries

In the natural language generation domain, various tasks exist, including language modeling, dialog response generation, and question-answer generation. Generally, NLG aims to generate a new sequence $\boldsymbol{x}_g = [x_{g,1}, \ldots, x_{g,n}]$ based on a provided context sequence $\boldsymbol{x}_c = [x_{c,1}, \ldots, x_{c,m}]$, where each x represents an individual token. The objective is to identify a model $f(\cdot)$ capable of generating an appropriate sequence using the given context: $f(\boldsymbol{x}_c) \rightarrow \boldsymbol{x}_g$. In cases like extractive answer generation, the answer is denoted by a pair of integer indices $(s, e) \in \mathbb{N}^2$, indicating the start and end positions of the answer span. Then the answer tokens $\boldsymbol{x}_a = [x_{c,s}, \cdots, x_{c,e}]$ can be located within the context sequence $\boldsymbol{x}_c$. It constitutes a part of $\boldsymbol{x}_g$ unless explicitly specified otherwise.

3.2 Model Architecture

VOLTA adheres to the VAE framework, where the encoder $f_{enc}(\cdot)$ and the decoder $f_{dec}(\cdot)$ are both Transformer models. Unlike Optimus (Li et al., 2020a), which utilizes BERT as the encoder and GPT-2 as the decoder, our model offers the simplicity of a shared backbone network between the encoder and decoder. Additionally, VOLTA can adapt encoder-decoder Transformers (Vaswani et al., 2017) seamlessly into VAE, leveraging their inherent encoder and decoder architecture.

uniform → equal probabilities across ranges → 6-side die

Latent variables from encoder. The encoder encodes the input sequence into multiple independent continuous or discrete latent variables, selecting the most suitable based on the input type. For instance, dialog responses and questions are aptly represented using continuous latent variables, aligning with their semantic nature. Conversely, discrete latent variables prove advantageous in modeling answer spans, aligning with their positions within the context. Specifically, latent variables can be calculated as follows:

context generate discrete probability

latent variable

$$\begin{aligned} \boldsymbol{h}_{enc} &= f_{enc}(\underbrace{\boldsymbol{x}_c}_{\text{context}}, \underbrace{\boldsymbol{x}_g}_{\text{generate}}), \\ &\boxed{\begin{array}{l} \mu_i, \sigma_i = \mathrm{FC}(\boldsymbol{h}_{enc}), \\ z_{g,i} \sim \mathcal{N}(\mu_i, \sigma_i^2), \end{array}} \quad \boxed{\begin{array}{l} \boldsymbol{\pi}_j = \mathrm{FC}(\boldsymbol{h}_{enc}), \\ z_{a,j} \sim \mathrm{Cat}(\boldsymbol{\pi}_j), \end{array}} \end{aligned} \tag{1}$$

continuous

where FC$(\cdot)$ is a single fully-connected layer and each instance has a distinct set of learnable parameters, indexing is omitted for simplicity; $\mathcal{N}(\cdot)$ is the Gaussian distribution with parameters μ_i and σ_i; Cat$(\cdot)$ is the categorical distribution whose parameters $\boldsymbol{\pi}_j$ represent the event probabilities of k categories. Back-propagation through the latent variables is achieved using the Gaussian distribution reparametrization trick (Wolpe and de Waal, 2019) for $\boldsymbol{z}_q = [z_{g,1}, z_{g,2}, z_{g,3}, \cdots]$ and Gumbel-Softmax (Maddison et al., 2017; Jang et al., 2017) reparametrization for $\boldsymbol{z}_a = [z_{a,1}, z_{a,2}, z_{a,3}, \cdots]$.

Latent codes. Supplementing the VAE latent variables, we incorporate InfoGAN-style latent codes (Chen et al., 2016) to infuse the model with input-independent variability. These latent codes come in two types: continuous latent codes, which can conform to either uniform distribution or Gaussian distribution, and discrete latent codes, which also adhere to categorical distribution:

continuous ← disc ←

$$\begin{aligned} \boldsymbol{c}_g &= [c_{g,1}, c_{g,2}, c_{g,3}, \cdots], \;\; c_{g,i} \sim \mathrm{Uni}(-1, 1), \\ \boldsymbol{c}_a &= [c_{a,1}, c_{a,2}, c_{a,3}, \cdots], \;\; c_{a,j} \sim \mathrm{Cat}(\boldsymbol{\rho}), \end{aligned} \tag{2}$$

where Uni$(\cdot)$ is the uniform distribution; the categorical distribution has parameters $\boldsymbol{\rho} = \frac{1}{k}\mathbf{1}$ that uses the same number of categories k as the discrete latent variables $\boldsymbol{z}_a$ because this compatibility is necessary as they will be concatenated together.

Cross-attention-based latent-space connection. Optimus (Li et al., 2020a) uses two channels to connect latent variables to the decoder: the 'embedding' channel involves a fully-connected layer to obtain a latent embedding, which is subsequently added to word embeddings. Meanwhile, the 'memory' channel generates latent embeddings for each Transformer block within the decoder. These latent embeddings are then concatenated with decoder hidden states as past information. The optimal performance is attained when both channels are utilized, albeit complicating the architecture.

In Transformers (Vaswani et al., 2017), the attention mechanism can take the form of self-attention or cross-attention. We introduce a unified and notably more effective cross-attention-based connection between the latent space and the decoder:

$$\begin{aligned} K_{\text{latent}} &= \mathrm{FC}([\boldsymbol{z}_g, \boldsymbol{c}_g]), \rightarrow \text{concatination} \\ V_{\text{latent}} &= \mathrm{FC}([\boldsymbol{z}_g, \boldsymbol{c}_g]), \\ \mathrm{Attention}(Q, K_{\text{latent}}, V_{\text{latent}}) & \\ &= \mathrm{softmax}\left(\frac{QK_{\text{latent}}^T}{\sqrt{d_k}}\right)V_{\text{latent}}. \end{aligned} \tag{3}$$

We facilitate the transmission of latent space information into the decoder using K_{latent} and V_{latent}, ? queried by the decoder via Q. In cases where the Transformer model lacks pretrained weights for cross-attention layers, such as decoder-only Transformers, we retain Optimus' connection method. However, we streamline it by incorporating a shared backbone for both the encoder and decoder.

Q comes from dec

Generation. VOLTA is trained in the typical autoregressive manner to predict subsequent tokens by considering the preceding tokens: in previous timesteps

$$\begin{aligned} \boldsymbol{h}_{g,t} &= f_{dec}(\underbrace{\boldsymbol{x}_c}_{\text{context}}, \underbrace{\boldsymbol{x}_{g,<t}}_{\text{generate}}, \underbrace{[\boldsymbol{z}_g, \boldsymbol{c}_g]}_{\text{latent code variable}}), \\ p(\boldsymbol{x}_g) &= \prod_{t=1}^{n} p(x_{g,t} \mid \boldsymbol{x}_c, \boldsymbol{x}_{g,<t}, [\boldsymbol{z}_g, \boldsymbol{c}_g]) \\ &= \prod_{t=1}^{n} \mathrm{softmax}(\mathrm{FC}(\boldsymbol{h}_{g,t})), \end{aligned} \tag{4}$$

where $\boldsymbol{x}_{g,<t}$ means the first $t-1$ tokens in $\boldsymbol{x}_g$.

The process for generating discrete data follows a similar approach but involves a distinct prediction head. Specifically, in the scenario of answer generation:

$$\begin{aligned} \boldsymbol{h}_a &= f_{dec}(\boldsymbol{x}_c, [\boldsymbol{z}_a, \boldsymbol{c}_a]), \\ p(s) &= \mathrm{softmax}(\mathrm{FC}(\boldsymbol{h}_{a,1:m})), \\ p(e) &= \mathrm{softmax}(\mathrm{FC}(\boldsymbol{h}_{a,1:m})), \\ s &= \operatorname*{arg\,max}_{s \in \{1,\cdots,m\}} p(s), \\ e &= \operatorname*{arg\,max}_{e \in \{1,\cdots,m\}} p(e), \\ \boldsymbol{x}_a &= [x_{c,s}, \cdots, x_{c,e}], \end{aligned} \tag{5}$$

where $\boldsymbol{h}_a$ denotes the hidden states obtained from the decoder; the subscript $1:m$ means slicing

	Specifications		**Quality**		**Diversity**				
Model	Type	Para.	EM	F1	Dist1	Dist2	Dist3	Dist4	S-BL ↓
GPT-2	TFM-Dec	124M	56.28	67.86	8.23	38.63	62.58	75.42	32.09
BART	TFM-Enc-Dec	139M	58.03	69.99	8.08	38.49	62.34	74.91	32.66
T5	TFM-Enc-Dec	222M	59.76	71.98	8.18	40.78	65.52	77.20	30.51
OPT	TFM-Dec	331M	58.57	70.40	7.88	38.51	63.80	76.55	29.97
HCVAE	VAE w/ LSTM	158M	61.81	73.68	7.00	33.47	57.24	71.68	32.66
Optimus	VAE w/ TFM	233M	58.05	69.55	8.05	40.27	66.63	79.88	29.28
VOLTA	VAE w/ TFM	124M	**65.56**	**77.31**	**8.32**	**40.84**	**68.05**	**82.64**	**28.34**

Table 2: Performance comparison on question-answer generation. *Abbreviations*: "HCVAE": Info-HCVAE; "Para.": Parameter Count; "Distk": Distinct-k; "S-BL": Self-BLEU; "TFM": Transformer; "Enc", "Dec": Encoder, Decoder; "↓" means lower is better.

an array from index 1 to m, which corresponds to the context tokens. This results in a generated answer $\boldsymbol{x}_a$, where s denotes the starting index and e denotes the ending index.

3.3 Training Objectives

Since the marginal likelihood $p(\boldsymbol{x})$ is intractable to compute, we approximate the true posterior $p(z \mid \boldsymbol{x})$ with $q(z \mid \boldsymbol{x})$ based on our encoder $f_{enc}(\cdot)$. Following the standard VAE formulation, we define the evidence lower bound (ELBO) as ELBO $= -\mathcal{L}_{\text{AE}}(\boldsymbol{x}) - \mathcal{L}_{\text{REG}}(\boldsymbol{z})$. Here, $\mathcal{L}_{\text{AE}}$ stands for Autoencdoer (AE) reconstruction loss and $\mathcal{L}_{\text{REG}}$ represents $D_{\text{KL}}(q(z \mid \boldsymbol{x}) \parallel p(z))$ for regularization.

Latent variable regularization loss. The KL divergence for regularizing the continuous or discrete latent variable is:

KL

$$\overset{\text{contin}}{\mathcal{L}_{\text{REG}}(z_g)} = \log\frac{\sigma'}{\sigma} + \frac{\sigma^2 + (\mu-\mu')^2}{2\sigma'^2} - \frac{1}{2},$$
$$\text{disc}\quad \mathcal{L}_{\text{REG}}(z_a) = \sum_{i=1}^{k} \pi_i \log\frac{\pi_i}{\pi'_i}, \qquad (6)$$

where $\mu, \sigma, \boldsymbol{\pi}$ follows Eq. (1); we assume that the priors $p(z_g)$ and $p(z_a)$ follow $\mathcal{N}(\mu', \sigma'^2)$ and Cat$(\boldsymbol{\pi}')$, respectively. In practice, μ', σ' and $\boldsymbol{\pi}'$ can be obtained by encoding only the context $\boldsymbol{x}_c$. The total $\mathcal{L}_{\text{REG}}$ is the mean over the latent variables. The derivations are in Appendix A.3, A.4.

Latent code VMIM loss. To prevent the model from ignoring the latent codes, we encourage it to recover the latent codes in the generation phase by optimizing the Variational Mutual Information Maximization (VMIM) objective (Chen et al., 2016):

Maximizing the mutual info between c, x'

$$I(\overset{\text{latent code}}{c}; \overset{x'}{\underline{f_{dec}(\boldsymbol{x}, [\boldsymbol{z}, c])}})$$
$$= \underbrace{H(c)}_{\text{entropy}} + \mathbb{E}_{\boldsymbol{x}'}\Big[D_{\text{KL}}(p(c' \mid \boldsymbol{x}') \parallel q(c' \mid \boldsymbol{x}')) + \mathbb{E}_{c'}\big[\log q(c' \mid \boldsymbol{x}')\big]\Big] \qquad (7)$$
$$\geq H(c) + \mathbb{E}_{\boldsymbol{x}'}\Big[\mathbb{E}_{c'}\big[\log q(c' \mid \boldsymbol{x}')\big]\Big]$$
$$\triangleq H(c) - \mathcal{L}_{\text{VMIM}}(c),$$

where $\boldsymbol{x}' \sim f_{dec}(\boldsymbol{x}, [\boldsymbol{z}, c])$; $c' \sim p(c \mid \boldsymbol{x}')$ is the recovered latent code. Because the posterior $p(c \mid \boldsymbol{x}')$ is difficult to obtain, an auxiliary distribution $q(c \mid \boldsymbol{x}')$ based on $f_{dec}(\cdot)$ is added to approximate it. The entropy $H(c)$ is a constant and thus excluded from $\mathcal{L}_{\text{VMIM}}(c)$. The derivation of this objective is included in Appendix A.5.

In our model, a fully-connected layer is added to the decoder for recovering each latent code c:

$$\theta = \text{FC}(f_{dec}(\boldsymbol{x}, [\boldsymbol{z}, c])),$$
$$\mathcal{L}_{\text{VMIM}}(c) = -\log p(c'; \theta), \qquad (8)$$

where the parameter θ depends on the distribution type of the corresponding latent code c. The total VMIM loss is the mean over the latent codes.

Overall objective. By Eq. (6)(8), the overall loss:

$$\mathcal{L}(x) = \mathcal{L}_{\text{AE}}(\boldsymbol{x}) + \beta\mathcal{L}_{\text{REG}}(\boldsymbol{z}) + \gamma\mathcal{L}_{\text{VMIM}}(\boldsymbol{c}), \quad (9)$$

where β, γ denote the coefficients used to adjust the loss weights; the Autoencoder reconstruction loss $\mathcal{L}_{\text{AE}}(\boldsymbol{x})$ corresponds to the standard cross-entropy loss employed for language modeling.

4 Experiments

4.1 Tasks and Datasets

We evaluate VOLTA against various baselines across six datasets, spanning three distinct NLG tasks:

Dataset	PTB			YELP			YAHOO			SNLI		
Model	PPL ↓	MI	AU	PPL ↓	MI	AU	PPL ↓	MI	AU	PPL ↓	MI	AU
M. A.	101.40	0.00	0	40.39	0.13	1	61.21	0.00	0	21.50	1.45	2
C. A.	108.81	1.27	5	-	-	-	66.93	2.77	4	23.67	3.60	5
SA-VAE	-	-	-	-	1.70	8	60.40	2.70	10	-	-	-
Aggressive	99.83	0.83	4	39.84	2.16	12	59.77	2.90	19	**21.16**	1.38	5
AE-BP	96.86	5.31	32	47.97	7.89	32	59.28	8.08	32	21.64	7.71	32
Optimus	51.39	0.02	0	27.63	0.02	0	29.35	0.04	0	66.58	9.20	32
VOLTA	**45.29**	**8.17**	32	**14.14**	**9.00**	32	**14.82**	**9.02**	32	25.69	**9.24**	32

Table 3: Performance comparison on language modeling tasks. Baseline results are obtained from Li et al. (2020a), excluding Optimus, which is not second-stage pretrained for a fair comparison. The maximum achievable AU is 32.

	Quality	**Diversity**	**Overall**
Model	Precision	Recall	F1
Seq2Seq	0.232	0.232	0.232
SeqGAN	0.270	0.270	0.270
CVAE	0.222	0.265	0.242
VHRED	0.341	0.278	0.306
VHCR	0.271	0.260	0.265
WAE	0.266	0.289	0.277
iVAE_{MI}	0.239	0.355	0.285
T5	0.321	0.321	0.321
Optimus	0.313	0.362	0.336
VOLTA	**0.373**	**0.401**	**0.387**

Table 4: Performance comparison on dialog response generation. Baseline results are from Li et al. (2020a) except T5.

- **Dialog response generation**: we utilize the DailyDialog dataset (Li et al., 2017b), comprising approximately 13K multi-turn conversations, averaging eight turns per dialog;

- **Question-answer generation** (QAG): we employ the SQuAD dataset (Rajpurkar et al., 2016), with approximately 100K question-answer pairs where the answers are extractive;

- **Language modeling**: four LM datasets: Penn Treebank (PTB) (Marcus et al., 1993), SNLI (Bowman et al., 2015), YELP, and YAHOO (Yang et al., 2017; He et al., 2019).

4.2 Implementation Details

We conduct experiments using two Transformer model variants: the decoder-only Transformer, leveraging the GPT-2 base model (Radford et al., 2019), and the encoder-decoder Transformer, utilizing the T5 base model (Raffel et al., 2020). With the decoder-only Transformer, since it comprises solely Transformer decoder blocks, we employ it as the shared backbone for both the encoder and decoder within VOLTA. In contrast, the encoder-decoder Transformer features distinct Transformer encoder and decoder, aligning conveniently with the VOLTA encoder and decoder structures. Throughout our experiments, all Transformer-based models load pretrained checkpoints from Huggingface [1], undergoing fine-tuning exclusively on the respective datasets. Unlike the approach in Optimus (Li et al., 2020a), no secondary-stage pretraining is executed.

Our model utilizes a default configuration comprising 32 Gaussian latent variables, along with 4 uniform latent codes. For extractive answers, we utilize 20 categorical latent variables and 5 categorical latent codes, all comprising 10 categories, as shown in Table 5. Training is performed over 10 epochs with a learning rate set to 5×10^{-5}. To address the KL vanishing issue (Bowman et al., 2016), we employ a linear annealing schedule for β (Li et al., 2020a). This includes an initial increasing phase covering the first 25% of training, ascending from 0 to a maximum value of 0.1 (Lee et al., 2020). Additionally, we set $\lambda = 1.0$ in the KL thresholding scheme (Li et al., 2019) for language modeling. We conducted the experiments on four TITAN V GPUs.

4.3 Metrics

While our focus lies in achieving diverse NLG, maintaining generative quality is paramount, as completely random sentences might achieve perfect diversity scores but lack meaningful content.

Generative quality. In dialog response generation, we evaluate generative quality using BLEU-

[1] https://huggingface.co/

	Configuration					Quality		Diversity				
Row	z_g	z_a	Var.	c_g	c_a	EM	F1	Dist1	Dist2	Dist3	Dist4	S-BL ↓
DFLT	32	20	✓	Rnd	Rnd	**65.56**	**77.31**	8.32	**40.84**	**68.05**	**82.64**	**28.34**
A	16	20	✓	Rnd	Rnd	63.00	75.45	8.18	38.73	63.54	76.94	34.81
B	64	20	✓	Rnd	Rnd	64.82	76.21	8.11	38.28	63.38	76.98	34.63
C	32	10	✓	Rnd	Rnd	61.73	74.32	**8.37**	39.66	65.77	80.36	30.75
D	32	40	✓	Rnd	Rnd	62.70	75.11	**8.37**	39.80	66.08	80.65	30.50
E	32	20	✗	Rnd	Rnd	45.72	57.65	5.12	20.39	31.53	37.75	76.04
F	32	20	✗	Fix	Rnd	46.71	58.49	4.44	16.40	24.51	28.77	84.19
G	32	20	✗	Rnd	Fix	46.37	58.37	4.50	16.49	24.45	28.59	84.66

Table 5: The ablation study of VOLTA's latent space on the QAG task. The orange text highlights the difference from the default configuration in row DFLT. *Abbreviations*: "Var.": variational(✓)/deterministic(✗) latent variables; "Rnd/Fix": random/fixed latent codes.

Context	The university is the major seat of the Congregation of Holy Cross (albeit not its official headquarters, which are in Rome). Its main seminary, Moreau Seminary, is located on the campus across St. Joseph lake from the Main Building ······
Q1	What catholic denomination is the university of new haven located in?
Q2	What is the main campus of moreau seminary?
Q3	What religious institution is located on the campus of moreau seminary?
Q4	What former retreat center is located near the grotto?
Q5	What religious denomination does the moreau seminary belong to?
Q6	What is the oldest building on campus?
Q7	What is the main seminary in the university of kansas?
Q8	What is the main seminary of the college?
Q9	What retreat center is located near the grotto?

Table 6: An example of latent variable interpolation.

precision (Papineni et al., 2002). For language modeling, we measure perplexity (PPL) and mutual information (MI) to assess quality. In question-answer generation, direct measurement against SQuAD reference sentences is not feasible because higher diversity in its nature means shifting the generated content away from these references. Instead, Zhang and Bansal (2019) proposed Question-Answering-based Evaluation (QAE), including three main steps: (a) use the QAG model to generate question-answer pairs for raw Wikipedia entries; (b) train a separate question-answering model on the generated QA pairs; (c) evaluate the QA model's performance on the SQuAD development set, using exact match (EM) and F1 metrics (Rajpurkar et al., 2016, 2018). Poor performance in step (c) reflects low quality of the generated QA pairs, indirectly assessing the QAG model's quality. BERT (Devlin et al., 2019) serves as the QA model in (b), and we utilize a QAMI loss to enhance QA pair relevance, akin to Info-HCVAE Lee et al. (2020).

Generative diversity. In dialog response generation, we assess diversity using BLEU-recall (Papineni et al., 2002) as the diversity measurement. In language modeling, we analyze the impact of VAE latent variables by tracking the number of active units (AU). Quantitatively measuring diversity in generated questions involves two metrics: Distinct-k (Li et al., 2016) and Self-BLEU (Zhu et al., 2018). Distinct-k calculates the ratio of distinct k-grams to the total number of generated words. Self-BLEU computes the average BLEU score (Papineni et al., 2002) for each sentence against all others, aiming for dissimilarity among generated sentences. We generate five QA pairs for each context.

4.4 Question-Answer Generation

We compare VOLTA with several state-of-the-art baselines on the question-answer generation task, as summarized in Table 2. We base it on GPT-2 to aim for the minimal model size, showcasing the efficiency of the VOLTA framework. The VAE components in VOLTA add a mere 0.46M parameters.

Context	Holy Cross Father **John Francis O'Hara** was elected vice-president in 1933 and president of Notre Dame in 1934. During his tenure at Notre Dame, he brought numerous refugee intellectuals to campus; ······
Q1	$c_q = -.8$ What was O'Hara's first name?
Q2	$c_q = -.6$ Who was elected vice president in 1933?
Q3	$c_q = -.0$ What was O'Hara's title prior to becoming vice president?
Q4	$c_q = +.4$ What was O'Hara's first title?
A	John Francis O'Hara

Context	During his 13 years the Irish won three national championships, had five undefeated seasons, won the Rose Bowl in 1925, and produced players such as George Gipp and the "Four Horsemen". ······	
A1 $c_a = 0$ five	**A2** $c_a = 3$ 1925	**A3** $c_a = 7$ three

Table 7: Continuous (c_q)/Discrete (c_a) latent code for varying question/answer generation.

The first four baseline models—GPT-2 (Radford et al., 2019), BART (Lewis et al., 2020), T5 (Raffel et al., 2020) and OPT (Zhang et al., 2022)—all rely on regular Transformer architectures, lacking the variational aspects found in VAE and thereby demonstrating lower generative diversity. Although Info-HCVAE (Lee et al., 2020) used VAE, it inherited LSTM's limitations. We therefore also adapt the Transformer-based VAE, Optimus (Li et al., 2020a), to question generation. Our VOLTA framework harnesses Transformer models' high capacity alongside the variability inherent in VAE and InfoGAN. It stands out by achieving superior diversity over all baselines while maintaining a relatively small model size.

4.5 Language Modeling

In language modeling, we employ T5-based VOLTA, comparing it with prior VAE approaches: M. A. (Bowman et al., 2016), C. A. (Fu et al., 2019), SA-VAE (Kim et al., 2018), Aggreesive Training (He et al., 2019), AE-BP (Li et al., 2019), and the Transformer-based VAE model, Optimus (Li et al., 2020a). Our findings revealed that when solely fine-tuned on LM datasets without prior extensive second-stage pretraining on large-scale datasets, the latent variables of the Optimus model (Li et al., 2020a) collapsed. The reason behind this could lie in Optimus employing two separate latent-space connection methods, which are challenging to optimize. On the contrary, VOLTA's unified cross-attention-based approach proves notably more stable.

4.6 Dialog Response Generation

We compare VOLTA with Optimus (Li et al., 2020a), the current state-of-the-art model, and several other baselines: Seq2Seq (Serban et al., 2016), SeqGAN (Li et al., 2017a), CVAE (Zhao et al., 2017), VHRED (Serban et al., 2017), VHCR (Subramanian et al., 2018), WAE (Gu et al., 2019), iVAE$_{\text{MI}}$ (Fang et al., 2019). VOLTA is based on T5 (Raffel et al., 2020) for dialog response generation, and we include T5 as a baseline to assess the impact of the VOLTA framework. We maintain VOLTA's generation process without incorporating a joint latent space and fusion regularization for history and response (Gao et al., 2019), enabling a more general approach compared to Optimus.

4.7 Ablation Study

To assess the impact of the cross-attention-based latent-space connection, we compare VOLTA with Optimus in language modeling and dialog response generation. Given that QAG uniquely involves both continuous and discrete latent variables/codes, we focus on ablating the latent space information specifically within this task. Hence, the impact of VOLTA's three primary components is as follows:

- **Cross-attention-based latent-space connection** (Table 3, 4): Optimus employs two distinct and intricate channels—embedding concatenation and summation—which pose challenges in optimization. In contrast, VOLTA's unified cross-attention-based approach is more stable. In language modeling, Optimus' latent variables even collapsed without their second-stage pretraining.

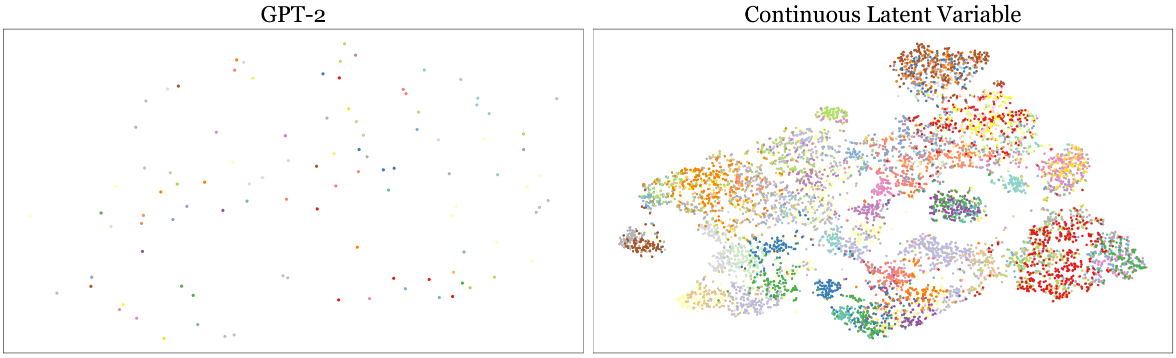

Figure 2: T-SNE visualization comparing question embeddings from GPT-2 with latent variable embeddings by VOLTA. Points of the same color depict embeddings from the identical context. VOLTA showcases diverse embeddings for each context, contrasting the deterministic nature of a vanilla LM.

- **Latent variables** (Table 5): Rows A-D show a general detriment when there is either an excess or a shortage of latent variables. Row E illustrates that when the latent variables become deterministic, the model essentially transforms into a conventional Autoencoder. Consequently, the performance experiences a significant decline, underscoring the critical role of the VAE framework.
- **Latent codes** (Table 5): Rows F, G depict a further decline in performance when we further fix the latent codes from Row E, underlining the latent codes' role in enhancing generative diversity.

4.8 Qualitative Analysis

Table 1 exemplifies a diverse instance of QAG achieved by the variational nature of VOLTA. Our model architecture facilitates two more methods to alter the generation process. One method involves interpolating latent variables, detailed in Table 6. The other method is centered on adjusting the InfoGAN-style latent codes, demonstrated in Table 7. In contrast to latent variables, latent codes are decoupled from the input context, affording the model more flexibility to explore the latent space.

To visualize the distribution of latent variables within the latent space, we utilize t-SNE (Van der Maaten and Hinton, 2008) to represent latent variable embeddings in a 2D space, comparing them with GPT-2 embeddings. Figure 2 illustrates that GPT-2 produces identical embeddings for a given context. Conversely, our model displays the ability to generate a cluster of diverse Gaussian latent variable points of the same color, subsequently decoded into a spectrum of distinct questions.

5 Conclusion

We present VOLTA, a framework merging the power of Transformers with the variability inherent in VAE and InfoGAN. Diverging from prior approaches, VOLTA introduces a novel cross-attention-based connection linking the latent space to the decoder, enhancing stability in optimization. This innovative architecture accommodates diverse Transformer types, including decoder-only or encoder-decoder architectures, and supports varying input types, be it continuous or discrete. Additionally, our framework incorporates InfoGAN-style latent codes, enabling input-independent variability, thereby further enriching generative diversity. Comprehensive experiments across six datasets spanning three distinct NLG tasks showcase VOLTA's significant enhancement in generative diversity while preserving quality.

6 Limitations

Given limited computational resources, we did not integrate LLM into the VOLTA framework, leaving this as a potential area for future exploration. As our model architecture is not confined to GPT-2 or T5, larger and more robust Transformer models could be employed to demonstrate its generalizability. Additionally, incorporating more NLG tasks and datasets could further reinforce our experimental results.

Acknowledgements

The research presented in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14222922, RGC GRF 2151185).

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*, pages 632–642. The 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 *CoNLL*, 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, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *ICLR*. OpenReview.net.

Jingcheng Deng, Liang Pang, Huawei Shen, and Xueqi Cheng. 2023. Regavae: A retrieval-augmented gaussian mixture variational auto-encoder for language modeling. In *EMNLP (Findings)*, pages 2500–2510. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. In *EMNLP/IJCNLP (1)*, pages 3944–3954. 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 *NAACL-HLT (1)*, pages 240–250. Association for Computational Linguistics.

Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Jointly optimizing diversity and relevance in neural response generation. In *NAACL-HLT (1)*, pages 1229–1238. Association for Computational Linguistics.

Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2019. Dialogwae: Multimodal response generation with conditional wasserstein autoencoder. In *ICLR (Poster)*. OpenReview.net.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. In *ICLR (Poster)*. OpenReview.net.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *NeurIPS*.

Dou Hu, Xiaolong Hou, Xiyang Du, Mengyuan Zhou, Lianxin Jiang, Yang Mo, and Xiaofeng Shi. 2022a. Varmae: Pre-training of variational masked autoencoder for domain-adaptive language understanding. In *EMNLP (Findings)*, pages 6276–6286. Association for Computational Linguistics.

Zhe Hu, Hou Pong Chan, Jiachen Liu, Xinyan Xiao, Hua Wu, and Lifu Huang. 2022b. Planet: Dynamic content planning in autoregressive transformers for long-form text generation. *arXiv preprint arXiv:2203.09100*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR (Poster)*. OpenReview.net.

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

Yoon Kim, Sam Wiseman, Andrew C. Miller, David A. Sontag, and Alexander M. Rush. 2018. Semi-amortized variational autoencoders. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 2683–2692. PMLR.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *ICLR*.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. 2020. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional vaes. In *ACL*, pages 208–224. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880. Association for Computational Linguistics.

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 *EMNLP/IJCNLP (1)*, pages 3601–3612. Association for Computational Linguistics.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020a. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *EMNLP (1)*, pages 4678–4699. Association for Computational Linguistics.

Jingjing Li, Zichao Li, Tao Ge, Irwin King, and Michael R Lyu. 2022. Text revision by on-the-fly representation optimization. pages 10956–10964.

Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael Lyu, and Irwin King. 2020b. Unsupervised text generation by learning from search. *Advances in Neural Information Processing Systems*, 33:10820–10831.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*, pages 110–119. The Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. In *EMNLP*, pages 2157–2169. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. In *IJCNLP(1)*, pages 986–995. Asian Federation of Natural Language Processing.

Ali Lotfi-Rezaabad and Sriram Vishwanath. 2020. Learning representations by maximizing mutual information in variational autoencoders. In *ISIT*, pages 2729–2734. IEEE.

Yueen Ma, Zixing Song, Xuming Hu, Jingjing Li, Yifei Zhang, and Irwin King. 2023. Graph component contrastive learning for concept relatedness estimation. In *AAAI*, pages 13362–13370. AAAI Press.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR (Poster)*. OpenReview.net.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguistics*, 19(2):313–330.

Gábor Melis, András György, and Phil Blunsom. 2022. Mutual information constraints for monte-carlo objectives to prevent posterior collapse especially in language modelling. *J. Mach. Learn. Res.*, 23:75:1–75:36.

OpenAI. 2023a. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2023b. Introducing chatgpt.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. ACL.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. SOLOIST: building task bots at scale with transfer learning and machine teaching. *Trans. Assoc. Comput. Linguistics*, 9:907–824.

Dong Qian and William K. Cheung. 2019. Enhancing variational autoencoders with mutual information neural estimation for text generation. In *EMNLP/IJCNLP (1)*, pages 4045–4055. Association for Computational Linguistics.

Zexuan Qiu, Jingjing Li, Shijue Huang, Wanjun Zhong, and Irwin King. 2024. Clongeval: A chinese benchmark for evaluating long-context large language models. *arXiv preprint arXiv:2403.03514*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL (2)*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392. The Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, pages 3776–3784. AAAI Press.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301. AAAI Press.

Zixing Song, Yifei Zhang, and Irwin King. 2023a. No change, no gain: Empowering graph neural networks with expected model change maximization for active learning. In *NeurIPS*.

Zixing Song, Yifei Zhang, and Irwin King. 2023b. Optimal block-wise asymmetric graph construction for graph-based semi-supervised learning. In *NeurIPS*.

Zixing Song, Yuji Zhang, and Irwin King. 2023c. Towards fair financial services for all: A temporal GNN approach for individual fairness on transaction networks. In *CIKM*, pages 2331–2341. ACM.

Sandeep Subramanian, Sai Rajeswar, Alessandro Sordoni, Adam Trischler, Aaron C. Courville, and Chris Pal. 2018. Towards text generation with adversarially learned neural outlines. In *NeurIPS*, pages 7562–7574.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Haoqin Tu, Zhongliang Yang, Jinshuai Yang, Siyu Zhang, and Yongfeng Huang. 2022. Adavae: Exploring adaptive gpt-2s in variational auto-encoders for language modeling. *CoRR*, abs/2205.05862.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *AAAI*, pages 7371–7379. AAAI Press.

Zach Wolpe and Alta de Waal. 2019. Autoencoding variational bayes for latent dirichlet allocation. In *FAIR*, volume 2540 of *CEUR Workshop Proceedings*, pages 25–36. CEUR-WS.org.

Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, and Bill Dolan. 2021. A controllable model of grounded response generation. In *AAAI*, pages 14085–14093. AAAI Press.

Xuerong Xiao, Swetava Ganguli, and Vipul Pandey. 2020. Vae-info-cgan: generating synthetic images by combining pixel-level and feature-level geospatial conditional inputs. In *IWCTS@SIGSPATIAL*, pages 1:1–1:10. ACM.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: controllable story generation with external knowledge using large-scale language models. In *EMNLP (1)*, pages 2831–2845. 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 *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 3881–3890. PMLR.

Fei Ye and Adrian G. Bors. 2021. Infovaegan: Learning joint interpretable representations by information maximization and maximum likelihood. In *ICIP*, pages 749–753. IEEE.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. In *EMNLP/IJCNLP (1)*, pages 2495–2509. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2019. Infovae: Balancing learning and inference in variational autoencoders. In *AAAI*, pages 5885–5892. AAAI Press.

Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL (1)*, pages 654–664. 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 *SIGIR*, pages 1097–1100. ACM.

A Appendix

A.1 Notations

Because we use VAE in this paper, our model is the composition of its encoder and decoder: $f = f_{enc} \circ f_{dec}$.

	Symbol	Description
Constant	m/n	The length of the context/question
	k	The number of categories in a categorical distribution
Variable	$\boldsymbol{x}$	Text sequence
	$\boldsymbol{z}/\boldsymbol{c}$	Latent variable/code vector
	z/c	A single latent variable/code
	$\Box_c/\Box_q/\Box_a$	Context/question/answer subscript
	$\Box_{\Box,i}$	Element index of a vector
	s/e	Answer span start/end token index
	$\Box'$	Generated content
Model	$f_{enc}(\cdot)$, $f_{dec}(\cdot)$	Encoder, decoder
	$\mathrm{FC}(\cdot)$	Single fully-connected layer
	$\mathcal{N}(\cdot)$	Gaussian distribution
	$\mathrm{Uni}(\cdot)$	Uniform distribution
	$\mathrm{Cat}(\cdot)$	Categorical distribution
	$[\cdots]$	Concatenation operation
	$\mathrm{CE}(\cdot)$	Cross-entropy loss

Table 8: Notations used in this paper.

A.2 Basic Definitions

Information is defined as:

$$I(X) = -\log P(X) = \log \frac{1}{P(X)}.$$

Entropy is defined as:

$$\begin{aligned} H(X) =& \mathbb{E}[I(X)] \\ =& \mathbb{E}[-\log(P(X))] \\ =& -\int p(x) \log p(x) \mathrm{d}x \\ H(X|Y) =& \mathbb{E}_{X,Y}[-\log \mathrm{P}(X|Y)] \\ =& -\int f(x,y) \log f(x|y) \mathrm{d}x\mathrm{d}y, \end{aligned}$$

where $p(x,y)$ is the probability mass function of a discrete distribution, whereas $f(x,y)$ is the probability density function of a continuous distribution.

Then mutual information is:

$$\begin{aligned} & I(X;Y) \\ =& D_{\mathrm{KL}}(P(X,Y) \parallel P(X)P(Y)) \\ =& \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \mathrm{d}x\mathrm{d}y \\ =& -\int p(x,y) \log p(y) \mathrm{d}x\mathrm{d}y \\ & + \int p(x,y) \log \frac{p(x,y)}{p(x)} \mathrm{d}x\mathrm{d}y \\ =& -\int p(y) \log p(y) \mathrm{d}y \\ & + \int p(x,y) \log p(y|x) \mathrm{d}x\mathrm{d}y \\ =& H(Y) - H(Y|X) \\ =& H(X) - H(X|Y), \end{aligned}$$

because Kullback–Leibler divergence is defined to be:

$$\begin{aligned} & D_{\mathrm{KL}}(Q \parallel P) \\ & \quad = H(Q,P) - H(Q) \\ & \quad = \mathbb{E}_Q[-\log \mathrm{P}(X)] - \mathbb{E}_Q[-\log \mathrm{Q}(X)] \\ & \quad = \int q(x) \log \frac{q(x)}{p(x)} \mathrm{d}x \\ & \quad \geq 0, \end{aligned}$$

where $H(Q,P)$ is the cross entropy of Q and P.

A.3 Optimus (Beta-VAE)

In Optimus (Li et al., 2020a; Kingma and Welling, 2014), we assume a normal distribution for a continuous latent variable:

$$\begin{aligned} f(x) =& \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \\ & \log f(x) \\ & \quad = -\log \sigma\sqrt{2\pi} - \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \\ & \quad = -\log \sigma - \frac{1}{2}\log 2\pi - \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \\ & \quad = -\frac{1}{2}\log \sigma^2 - \frac{1}{2}\log 2\pi - \frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2. \end{aligned}$$

We want $q(z|x) = N(\mu_q, \sigma_q^2)$ and the prior, $p(z) = N(\mu_p, \sigma_p^2) = N(0, 1)$, to be close

$$\begin{aligned}
&D_{\mathrm{KL}}(Q \parallel P) \\
&= -\int q(z) \log p(z) dz + \int q(z) \log q(z) dz \\
&= \left(\frac{1}{2}(\log 2\pi\sigma_p^2) + \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} \right) \\
&\quad - \frac{1}{2}(1 + \log 2\pi\sigma_q^2) \\
&= \frac{1}{2}(\log \frac{\sigma_p^2}{\sigma_q^2}) + \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} - \frac{1}{2} \\
&= \frac{1}{2} \log \left(\frac{\sigma_p}{\sigma_q} \right)^2 + \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} - \frac{1}{2}
\end{aligned}$$

The mutual information between z and $z|x$ is

$$I(z, x) = H(z) - H(z|x),$$

where the negative entropy for normal distribution is (n_z is the dimension of latent variable z):

$$\begin{aligned}
-H(z|x) &= \mathbb{E}_{Q(z|x)}[\log(Q(z|x))] \\
&= -\int q(z) \log q(z) \mathrm{d}z \\
&= -\frac{1}{2}(1 + \log 2\pi\sigma_q^2) \\
&= -\frac{1}{2}(1 + \log 2\pi + \log \sigma_q^2) \\
&= -\frac{1}{2} \log 2\pi - \frac{1}{2}(1 + \log \sigma_q^2)
\end{aligned}$$

$$\begin{aligned}
&H(z) = \mathbb{E}_{q(z)}[-\log q(z)] \\
&= -\int q(z) \left(\log \sigma_q \sqrt{2\pi} + \frac{1}{2} \left(\frac{z - \mu_q}{\sigma_q} \right)^2 \right) \mathrm{d}x \\
&= -\int q(z) \log \sigma_q \sqrt{2\pi} \mathrm{d}x \\
&\quad - \int q(z) \frac{1}{2} \left(\frac{z - \mu_q}{\sigma_q} \right)^2 \mathrm{d}x \\
&= -\mathbb{E}_{q(z)}[\log \sigma_q \sqrt{2\pi}] - \mathbb{E}_{q(z)} \left[\frac{1}{2} \left(\frac{z - \mu_q}{\sigma_q} \right)^2 \right] \\
&= -\log \sigma_q \sqrt{2\pi} - \mathbb{E}_{q(z)} \left[\frac{1}{2} \left(\frac{z - \mu_q}{\sigma_q} \right)^2 \right] \\
&= -\log \sigma_q \sqrt{2\pi} - \frac{1}{2} \left(\frac{\mathbb{E}_{q(z)} \left[(z - \mu_q)^2 \right]}{\sigma_q^2} \right) \\
&= -\frac{1}{2} \log \sigma_q^2 - \frac{1}{2} \log 2\pi - \frac{1}{2} \frac{(z - \mu_q)^2}{\sigma_q^2},
\end{aligned}$$

where $\mathbb{E}_{q(z)} \left[(z - \mu_q)^2 \right]$ is simply the deviation of a single sample z from the mean μ_q.

A.4 Info-HCVAE

According to Info-HCVAE (Lee et al., 2020), some inputs are better suited to be encoded into discrete latent variables. In this case, we can make use of the categorical distribution:

$$f(x = i \mid \boldsymbol{p}) = p_i,$$

where the event probabilities $\boldsymbol{p} = (p_1, \ldots, p_k)$ and $\sum_{i=1}^k p_i = 1$; $k > 0$ is the number of categories.

The Gumbel-Softmax distribution enables back-propagation through discrete distributions. The Gumbel distribution is:

$$\text{Gumbel}(\mu, \beta) = f(x; \mu, \beta) = \frac{1}{\beta} e^{-(z + e^{-z})},$$

where $z = \frac{x - \mu}{\beta}$.

To sample a category from the categorical distribution using the Gumbel-Max re-parametrization trick, one can follow:

$$\arg\max_i (G_i + \log p_i),$$

where $G_i \sim$ Gumbel$(0, 1)$. $\arg\max$ can be made differentiable by approximating it with the softmax function:

$$y_i = \frac{\exp((G_i + \log p_i)/\tau)}{\sum_j \exp((G_j + \log p_j)/\tau)},$$

Given two categorical distributions P and Q, parameterized by $\boldsymbol{p}$ and $\boldsymbol{q}$, respectively, the KL divergence between them is:

$$D_{\mathrm{KL}}(Q \parallel P) = \sum_{i=1}^k q_i \log \frac{q_i}{p_i}.$$

A.5 InfoGAN

The input noise z is passed into the generator along with the latent code c: $G(z, c)$, where z is concatenated with c. Because the generator can simply ignore the latent code c, InfoGAN (Chen et al., 2016) adds Variational Mutual Information Maximization (VMIM) to maintain the mutual information between generated sample $x \sim G(z, c)$ and latent code c:

$$
\begin{aligned}
& I(c; G(z,c)) \\
=& H(c) - H(c|G(z,c)) \\
=& H(c) + \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log P(c'|x)]] \\
=& H(c) + \mathbb{E}_{x\sim G(z,c)}\Big[\sum_{c'} p(c'|x)\log p(c'|x)\Big] \\
=& H(c) + \mathbb{E}_{x\sim G(z,c)}\Big[\sum_{c'} p(c'|x)(\log \frac{p(c'|x)}{q(c'|x)} \\
& \qquad + \log q(c'|x))\Big] \\
=& H(c) + \mathbb{E}_{x\sim G(z,c)}\Big[\sum_{c'} p(c'|x)\log \frac{p(c'|x)}{q(c'|x)} \\
& \qquad + \sum_{c'} p(c'|x)\log q(c'|x)\Big] \\
=& H(c) + \mathbb{E}_{x\sim G(z,c)}\big[D_{\mathrm{KL}}(P(\cdot|x) \parallel Q(\cdot|x)) \\
& \qquad + \mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]\big] \\
\geq& H(c) + \mathbb{E}_{x\sim G(z,c)}\left[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]\right],
\end{aligned}
$$

Because the posterior $P(c|x)$ is hard to obtain, an auxiliary distribution $Q(c|x)$ is added to approximate $P(c|x)$, where Q is a neural network. In practice, the entropy of latent codes $H(c)$ is treated as a constant and omitted in the InfoGAN objective.

A.6 InfoVAE and InfoMax-VAE

The evidence lower bound (ELBO) of regular VAE is

$$
\begin{aligned}
& \mathcal{L}_{\mathrm{ELBO}}(x) \\
=& \mathcal{L}_{\mathrm{AE}}(x) + \mathcal{L}_{\mathrm{REG}}(x) \\
=& \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z)) \\
\leq& \log p_\theta(x).
\end{aligned}
$$

InfoVAE (Zhao et al., 2019) and InfoMax-VAE (Lotfi-Rezaabad and Vishwanath, 2020) add mutual information to the loss:

$$
\begin{aligned}
\mathcal{L}_{\mathrm{ELBO}}(x) =& \mathcal{L}_{\mathrm{AE}}(x) + \beta\mathcal{L}_{\mathrm{REG}}(x) + \alpha I_q(x;z) \\
=& \mathbb{E}_{p_D(x)}[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]] \\
& - \beta\mathbb{E}_{p_D(x)} D_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z)) \\
& - \alpha D(q_\phi(x;z) \parallel q(x)q_\phi(z)),
\end{aligned}
$$

Because $D(q_\phi(x;z) \parallel q(x)q_\phi(z))$ is usually intractable; thus, it can be approximated with any one of the following:

- KL divergence
- f-divergence (InfoMax)
- Donsker-Varadhan dual representation (InfoMax)
- Jensen Shannon divergence (AAE)
- Stein Variational Gradient
- Maximum-Mean Discrepancy

A.7 QA mutual information loss

We want to enforce the mutual information (QAMI) between the generated QA pair. Following Info-HCVAE (Lee et al., 2020), we base this QAMI objective on Jensen-Shannon Divergence, which uses a bilinear layer on top of the decoder to classify whether the question and answer is a true pair:

$$
\begin{aligned}
& \boldsymbol{h}_q = \overline{\boldsymbol{h}}_{q,m+1:m+n} \quad \boldsymbol{h}_a = \overline{\boldsymbol{h}}_{a,1:m} \\
& g(q,a) = \sigma(\boldsymbol{h}_q{}^T \boldsymbol{W} \boldsymbol{h}_a) \\
& I(q,a) \geq \mathbb{E}[\log g(q,a)] + 1/2\mathbb{E}[\log(1-g(\tilde{q},a))] \\
& \qquad\qquad + 1/2\mathbb{E}[\log(1-g(q,\tilde{a}))] \\
& \qquad = -\mathcal{L}_{\mathrm{QAMI}}(\boldsymbol{x}), \qquad (10)
\end{aligned}
$$

where the question and answer embeddings, $\boldsymbol{h}_q$ and $\boldsymbol{h}_a$, are the average of their contextualized token embeddings; $\boldsymbol{W}$ is the parameter matrix of the bilinear layer $g(\cdot)$; $\tilde{q}/\tilde{a}$ is a negative question/answer sample; $\sigma(\cdot)$ is the activation function.

A.8 Examples

We provide a few more examples of latent code control.

	Knute Rockne became head coach in 1918. Under Rockne, the Irish would post a record of 105 wins, 12 losses, and five ties ···	
Q1	$c_q = -1$	How many wins did Knute Rockne post?
Q2	$c_q = -.9$	How many wins did Knute Rockne have?
Q3	$c_q = -.5$	How many wins did the Irish post in 1918?
Q4	$c_q = +.9$	How many wins did the Irish post a record of in 1918?
	The Lobund Institute grew out of pioneering research in germ-free-life which began in 1928 ···	
Q1	$c_q = -1$	When did the institute begin research on germ free-life?
Q2	$c_q = -.8$	When did research in animal and plant life begin?
Q3	$c_q = -.5$	When did Lobund begin research on germ?
Q4	$c_q = -.1$	When did the Lobund Institute begin its research?
Q5	$c_q = +.5$	When did research in germ free-life begin?

Table 9: Examples of latent codes. Answer in blue. The latent code seems to control how specific the question is.