June 28th, 2021

Master's Thesis

MSc Artificial Intelligence

Taking a step back: assessing the TransformerVAE as a latent variable model first

---

Author

Claartje Barkhof

Academic Supervisors

David Stap

Dr. Wilker Ferreira Aziz

Examiner

Dr. Vlad Niculae

External Supervisors

Joris Baan | *DPG Media*

Lucas de Haas | *DPG Media*

Language Technology Lab

& Institute of Logic, Language and Computation

Universiteit van Amsterdam

Abstract

Deep generative latent variable modelling conceptually forms an exciting perspective on representation learning, by defining a hierarchical process in which latent variables are used to explain regularities in observed data. The resulting representations may therefore uncover high-level structures that are associated with intricate patterns in data space while also having the potential to generalise outside of the empirical data distribution. A Variational Autoencoder (VAE) is a probabilistic framework that prescribes a way how to learn such a model from (big) data according to the principles of variational inference, leveraging the power of deep neural networks to approximate complex probability distributions (Kingma & Welling, 2014). Because the qualitative goals of representation learning are not inherently aligned with the numerical goals of learning a latent variable model, optimisation in practice may lead to solutions where the latent representations are ignored by the generative model. This issue is known as *posterior collapse* (Bowman et al., 2016) and is especially likely to occur in the context of powerful generator networks, or *strong decoders* (Bowman et al., 2016; Alemi et al., 2018a).

The field of representation learning in the context of language, which will be the topic of this thesis, has taken flight in an orthogonal direction: designing ever-larger Transformer architectures (Vaswani et al., 2017) that have shown to be effective in a wide variety of tasks, but often make for a form of black-box natural language processing (NLP) that does not exhibit the aforementioned properties generative latent variable models naturally possess. C. Li et al. (2020) have recently made an attempt to unify these two lines of research in a new architectural class of the VAE to model language that we refer to as the TransformerVAE. In this thesis, we take a step back and present a mode of analysis that deviates from what is common in NLP and aim at explicitly evaluating what we argue should be the very goal of this new line of research: learning statistically healthy models that expose a meaningful organisation of the latent space in the context of (very) powerful density estimators as large pre-trained Transformer networks are. In the process of doing so, we will zoom in with an information theoretic lens to arrive at the conclusion there is an axis of variation (i.e. *marginal KL*) not accounted for in a well-established rate-distortion view on VAEs (Alemi et al., 2018a) that is directly relevant to this goal. We analyse existing optimisation techniques that target a specific rate in the hope to circumvent posterior collapse with regards to this quantity and find notable differences that lead to practical recommendations. Additionally, we translate this analytical view into consequences for optimisation and conceptually identify potential pathological optimisation directions concerning marginal KL that pose a hazard especially when aiming for solutions with high rate.

Acknowledgements

First, I would like to express my gratitude towards David Stap for his role as supervisor and all the good guidance he has provided me with on writing this thesis throughout this academic year. Secondly, I would like to to thank Dr. Wilker Aziz for his willingness to be involved as an additional supervisor, all the time and considerate effort he spent in this collaboration and for sparking my academic interest even further.

Furthermore, I would like to thank DPG Media for the opportunity of the internship this thesis is the result of and in particular Joris Baan and Lucas de Haas for the curiosity and interest they showed in the topic of my thesis and the time they invested in the many nice meetings we had this year.

Lastly, I would like to thank Vlad Niculae for taking on the role of examiner.

Contents

Chapter 1

Introduction

Representation learning is a core ingredient of the success of modern day machine learning. It revolves around automatically extracting features from data. The representations can be used directly in an algorithm to perform a task or in a less direct way in a subsequent learning setting. In the case of language, which will be the focus of this thesis, direct usage of representations for a task could be retrieving relevant documents given a query or classifying the topic of a text. When representations serve as a starting point in a downstream task, a practice called *transfer learning*, they have the potential to ease the accessibility to relevant information present in the data for the task at hand. This can increase the efficiency of the subsequent learning task and, in effect, decrease the resources needed. Although these representations have been shown to be of crucial importance for the performance of machine learning algorithms (Bengio, Courville, & Vincent, 2013), the question of what a "good" representation should comprise of and how to efficiently learn it, are non-trivial questions to answer and the subject of ongoing research.

1.1 Generative latent variable models of language

In a probabilistic setting, representation learning can conceptually be viewed as learning a generative latent variable model. A generative latent variable model is a statistical framework for describing the hierarchical interaction between unobserved variables Z and observed variables X. Representations are intuitively modelled by the latent variables, while the data are modelled by the observed variables. Generation is assumed to follow the directed path of sampling a latent variable from a prior distribution and subsequently, given that information, determining a distribution over the data space. Inference is defined as the opposite directed path: from the observed to the unobserved variable. In general, we need only to define and fix one such direction, as the other follows by laws of probability calculus (which is the classical interpretation of inference). These directed stochastic dependencies are shown in Figure 1.1. For the purpose of representation learning, the latent variables may govern abstract and high level features of the data and thus typically live in a lower dimensional space than the observed variables. As Bengio et al. (2013) formulate it, we want the latent variables to capture the explanatory or generative factors of variation of the data. They argue this is important for robust usage of representations in subsequent learning settings.

A Variational Autoencoder (VAE) (Kingma & Welling, 2014) is a framework to learn a generative latent variable model at scale leveraging information from large datasets. It is

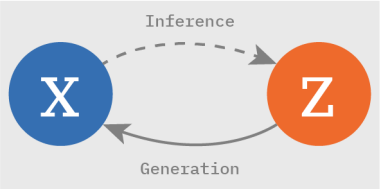

FIGURE 1.1: Directed stochastic dependencies in a generative latent variable model

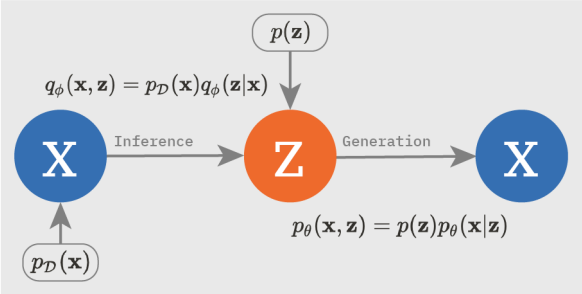

FIGURE 1.2: Two factorisations of the joint distribution between observed and unobserved variables in a Variational Autoencoder

optimised using the principles of variational inference (Jordan & Saul, 1999), using an approximation $q_\phi(\mathbf{z}|\mathbf{x})$ of the otherwise intractable true generative posterior $p_\theta(\mathbf{z}|\mathbf{x})$. With this approximate posterior, it models the joint distribution of observed and unobserved variables in two ways and optimises a balance between the two: $q_\phi(\mathbf{x},\mathbf{z}) = p_\mathcal{D}(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x},\mathbf{z}) = p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$. On the one hand, the factorisation in the approximate posterior (or *inference*) direction is grounded by instantiating the marginal data distribution empirically by means of a dataset, denoted with $p_\mathcal{D}(\mathbf{x})$. On the other hand, the factorisation in the generative direction is grounded by a predefined distribution representing our prior belief over the latent variables, denoted with $p(\mathbf{z})$. The conditionals $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ are modelled by deep neural networks. More details on this optimisation procedure can be found in Section 2.2.1.

Because VAEs model both the inference view and generative view explicitly, we have the means to efficiently evaluate the distribution over $\mathbf{x}$ given $\mathbf{z}$ and approximate the induced distribution over $\mathbf{z}$ given $\mathbf{x}$. In the context of language, this entails that a sequence of text $\mathbf{x}$ is compressed into one stochastic representation $\mathbf{z}$ that potentially captures global characteristics that can be mapped back to the language space of $\mathbf{x}$ via the generative path. Global, or high-level, characteristics of language may for example include topic, sentiment and stylistic features. This is notably different from a regular language model, in which there exists no directly accessible, compact and stochastic global representation. In regular language models, the variation of generation is directly governed by the conditional information, which can be previous output of the model or conditional input presented to the model artificially. This causes the variation to be rather local and high frequency in nature. A global representation originating from a VAE, on the contrary, potentially allows for controlling variation on a higher level (X. Chen et al., 2017).

Apart from being high level, the representations inferred by a VAE can exploit structure in the latent space in an interesting way. As opposed to a regular autoencoder (AE), a VAE is a directed probabilistic model which encodes data samples not as point estimates but with

posterior distributions that, in case of continuous latent variables, define soft regions for the latent representation. This smoothness makes that neighbourhood in the latent space can be associated with complex patterns in the data space (Pelsmaeker & Aziz, 2019). It also makes manipulation of the latent representation possible. Two points in representation space can exchange high-level features and induce a complex, smooth and credible change in the data space. To put another way, meaningful changes made in the latent space will result in points in the data space that lay on the data manifold of natural language (J. Zhao, Kim, Zhang, Rush, & LeCun, 2018). This is notably different from regular language models in which the potentially learned data manifold is not directly available to us and can only be interacted with via the data space. And, note that the fact that we have a way to approximate the inference direction is useful, as this gives us an idea of what the true posterior might look like, which is of great interest. The posterior tells us what regularities in the data space the generative model maps to the lower dimensional latent manifold to best explain the observed data, which is exactly what we are after.

Practically, a VAE employs deep neural networks for inference and generation: the encoder network to model $q_\phi(\mathbf{z}|\mathbf{x})$ and decoder network to model $p_\theta(\mathbf{x}|\mathbf{z})$, respectively. As language is inherently sequential, architectures that deal with this sequential nature through recurrence, such as the LSTM (Hochreiter & Urgen Schmidhuber, 1997), have been a popular choice to use as the encoder and decoder networks of VAE language models (Dieng, Kim, Rush, & Blei, 2019; Fang, Li, Gao, Dong, & Chen, 2019; Fu et al., 2019; B. Li, He, Neubig, Berg-Kirkpatrick, & Yang, 2020). Although a recurrent architecture in theory is an adequate fit, the field of natural language processing has moved beyond recurrent architectures to replace them with Transformer architectures (Vaswani et al., 2017). These architectures rely on the self-attention mechanism and positional embeddings, rather than recurrence, to model dependencies between time steps. The self-attention mechanism allows for using a great deal of parallelisation, causing the Transformer to win the proverbial hardware lottery (Hooker, 2020) given modern day parallel processing units. Transformer-based architectures have been shown to be successful in generative language modelling (Peters et al., 2018) and natural language understanding (Devlin, Chang, Lee, & Toutanova, 2019) and lend themselves well for transfer learning, including sequence-to-sequence modelling (Rothe, Narayan, & Severyn, 2020). A natural development is thus to incorporate the Transformer architecture into a VAE to model language. To our knowledge, C. Li et al. (2020) were among the first the first to use large, pre-trained transformer-based language models in a VAE setting.

1.2 Posterior collapse

As pointed out before, VAEs theoretically form an exciting direction in representation learning: combining the statistical transparency of latent variable models with the power of deep learning. In practice, however, research has shown that the goal of representation learning is often undermined when learning such a model from data (Bowman et al., 2016). Although the goal is to learn an practically useful dependency between the latent and observed variables that may be used by the generative model, researchers find that the generative model tends to ignore the latent variables altogether (Alemi et al., 2018a; Bowman et al., 2016; Fu et al., 2019; B. Li et al., 2020). The model reduces to a regular language model in this case. In statistical terms, this leads to a reduction of the joint distribution in the generative direction to a factorisation of independent marginals and, thus, a posterior that collapses to a prior

(hence the name *posterior collapse*).

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \;\rightarrow\; p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x})p(\mathbf{z}) \;\rightarrow\; p_\theta(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}) \tag{1.1}$$

It is important to realise that posterior collapse is rather the result of an ill-positioned idea of framing representation learning as learning a latent variable model than failure of the VAE framework from a optimisation point of view (Alemi et al., 2018a; X. Chen et al., 2017; S. Zhao, Song, & Ermon, 2018). As a VAE is merely concerned with balancing two views of the joint distribution between $\mathbf{x}$ and $\mathbf{z}$, a balance between joints in which the latent and observed variables are independent may be a perfectly valid, yet uninteresting, solution. An important factor in causing posterior collapse, is the ability of the generative network, or decoder $p_\theta(\mathbf{x}|\mathbf{z})$, to model the data distribution without making use of the latent representation (Pelsmaeker & Aziz, 2019; Alemi et al., 2018a; Kim, Wiseman, & Rush, 2018; Bowman et al., 2016; Fu et al., 2019; S. Zhao, Song, & Ermon, 2017). Since any joint distribution over a sequence of text can be modelled with an auto-regressive factorisation (as is typical in decoders) together with the fact that our choice of parametric family (i.e. the categorical distribution) is the maximally expressive choice of conditional for language within this factorisation, we have reason to believe the decoder can actually come a long way in modelling the data distribution without making use of the latent variables. We will expand on this argument further in Section 2.3.1. From this perspective, posterior collapse is in fact not even so surprising to occur. And thus, it remains an open research question how to incorporate powerful density estimators such as auto-regressive language models in a VAE setting while also meeting the expectations of representation learning (i.e. how to solve the *strong decoder problem*). A more thorough review of posterior collapse can be found in Section 2.3.

1.3 Problem statement

We find ourselves at a point where two interesting lines of research coincide. On the one hand we see advances in density estimation in the context of language modelling using ever-larger Transformer networks and on the other hand the use of deep generative latent variable models as an exciting perspective on representation learning. The previously mentioned work of C. Li et al. (2020) can be seen as a starting point of this conjunction of introducing large pre-trained Transformer networks into the VAE setting. We will refer to this model class as the TransformerVAE from now on. The authors showcase the performance of this new model class along axes that are commonly found in NLP research. They report on global statistics (such as perplexity values), performance on downstream tasks and present anecdotal evidence in the form of text samples. In this thesis, we aim to take a step back and analyse this new model class as a generative latent variable model first, with which we mean to find a good balance between the following four goals:

1. **Capturing the data distribution** We would like to learn a generative model that captures the data generative process, in the sense that it explains the observed data well.

2. **Latent space structure** We would like this generative model to exploit a dependency between $\mathbf{x}$ and $\mathbf{z}$. For this we need to overcome the state of a collapsed posterior. Ideally, we would like this dependency to be a meaningful and practically useful one.

3. **Consistency** We would like our VAE framework to achieve these goals in a statistically healthy way. That is to balance the two views of the joint as well as it can, where in the ideal setting $q_\phi(\mathbf{x}, \mathbf{z}) \approx p_\theta(\mathbf{x}, \mathbf{z})$.

4. **Accurate approximate inference** As we are to make use of strong decoders, there is attention to be paid to the decoder not overcompensating for the potentially inefficient encoder we have. If there is a dependency between $\mathbf{x}$ and $\mathbf{z}$ utilised by the decoder, we would our encoder to serve this mechanism as efficiently as possible and thus perform accurate approximate inference.

It is important to note that these goals are partially overlapping. For example, if the model is perfectly consistent (goal 3), our inference must be accurate as well (goal 4). But, we think it is useful to separate them as they represent trade-offs made in VAEs in which these goals are not satisfied to perfection, which is what we are likely dealing with in practice. Regarding goal 4 specifically: we have separated this goal from goal 3 as we will see that some objectives found in literature deviate from directly optimising for 3, which can have qualitative consequences for goal 4. Moreover, these goals might be in tension with each other, so focusing on a subset only might give a deceiving view of the quality of learned VAEs. For instance, we know that consistency (goal 3), can be achieved well without exploiting structure in the latent space (goal 2). This is the case for a VAE in which the posterior has collapsed.

Metrics as reported by C. Li et al. (2020) only partly cover these goals. Perplexity values averaged for a (test) data set—in case *estimated* in the correct way (Logan IV, Gardner, & Singh, 2020)—can give an average indication of goal 1, so does not explicitly reveal trade-offs made within the VAE regarding the other goals. Performance on downstream tasks can only *indirectly* be tied to the second goal. Additionally, if our main goal is to achieve high performance on downstream tasks, there might be more efficient ways to go about this. Self-supervised methods, for example, have been shown to be very effective at this: e.g. Lan et al. (2019), Devlin et al. (2019) and Raffel et al. (2019). Text samples, although nice to inspect, do not give a reliable view of the competency of the model as they are hard to assess (harder than for example images) and reporting enough of them would take up too much space.

In this thesis we will present an analysis that is meant to give an alternative and *complementary* view on assessing these goals. Our analysis will have an emphasis on goal 4, as we find this goal is underexposed in current literature and especially important in the context of strong decoders. We stress it is important to make an attempt at such an alternative evaluation, because if one of these goals is largely neglected it defeats the purpose of using a generative latent variable model (or more concretely a VAE) in the first place.

1.4 Research structure and contributions

We will continue to describe the structure of this thesis and highlight the contributions made. It should be pointed out that this thesis is in no way aiming at improving upon the work of C. Li et al. (2020) in terms of the metrics they present. The goal is rather to explore an alternative way of understanding its inner workings as a new architectural class of VAEs to model language.

1. **Demonstrating the information preference problem: a TransformerVAE is a *very* strong decoder.** We start our research by demonstrating that the TransformerVAE is

an instance of a strong decoder VAE. Although it is intuitive that a Transformer meets the criteria of a strong decoder and on its own it is not a novel insight that there is a key role for strong decoders in posterior collapse, we still think this demonstration is of added value. First, we think it is important to stress the existence of this problem in this model class as C. Li et al. (2020) leave this implicit. As it is precisely the goal of this new line of research to use powerful decoders in the context of a VAE, failing to address this issue gets this line of research off on the wrong foot. Secondly, it is an important realisation in order to assess the intuitive and relative effectiveness of different techniques to combat posterior collapse.

2. **A missing axis in the information theoretic view of VAEs: the quality of approximate posterior inference.** We continue our analysis with exploring an information theoretic view to arrive at a novel insight that there is a degree of freedom not accounted for in the rate-distortion perspective on VAEs that has been widely adopted in recent years of VAE research (Alemi et al., 2018b). We reason that this axis of evaluation (i.e. *marginal KL*) is directly relevant to the aforementioned goals: assessing the quality of the learned inference model. We analyse existing strategies to combat posterior collapse along this proposed axis and show surprising differences between those techniques with practical implications. Additionally, we present a conceptual outline of what the consequences of this quantity are for optimisation and identify potential pathological optimisation directions that current views in literature do not account for.

3. **On architectural inductive biases.** In the last part of this research we will assess the TransformerVAE from a architectural point of view and investigate the relation between architectural design and what information is likely to be encoded it the latent representation. We speculate that the design of the TransformerVAE as proposed by C. Li et al. (2020), or more specifically the *memory mechanism*, biases towards encoding local information. We argue that this analysis is insightful to conduct when experimenting with designing new architectural flavours of the TransformerVAE.

1.4.1 Outline thesis

The outline of this thesis will be as follows. In Chapter 2, we will give background information on language models, how to extend them to deep generative language models and give a review on posterior collapse and methods found in literature to counteract it. In Chapter 3 we will describe the implementation of the TransformerVAE we use for all experiments conducted for this thesis, that closely follows the work of C. Li et al. (2020) and report on training details to make this thesis computationally feasible. In Chapter 4 we will present our main analysis and deal with research point 1. in Section 4.1, point 2. in Section 4.2 and point 3. in Section 4.3. Finally, Chapter 5 comprises of a conclusion and hints at directions for future research.

Chapter 2

Background & related work

In this chapter we will provide a theoretical basis on which this thesis research builds and introduce some of the most important related work. This includes a review of the following topics: deep neural language models (Section 2.1), the theoretical extension of those into deep generative latent variable language models (Section 2.2) and, finally, posterior collapse in Section 2.3 with a literature review of the most prevalent techniques proposed to counteract it.

2.1 Deep neural language models

A typical way of learning a parameterised model that explains a set of observed data is via maximum likelihood estimation (MLE). In this learning paradigm, we assume the data to be independent and identically distributed (i.i.d.), so that we can evaluate the (log) likelihood of our model given the data with some fixed set of parameters θ by factorising over the data points. Under the MLE criterion, the optimal setting of the parameters $\theta^\star$ is the one that maximises this likelihood for some dataset $\mathbf{X} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}\}$:

$$\theta^\star = \underset{\theta}{\operatorname{argmax}}\ p_\theta(\mathbf{X}) = \underset{\theta}{\operatorname{argmax}}\ \log \prod_{i=1}^{N} p_\theta(\mathbf{x}^{(i)}) = \underset{\theta}{\operatorname{argmax}}\ \sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)}) \tag{2.1}$$

As can be seen from Equation 2.1, this objective summons us to assess the probability a model assigns to a datum. In the case of language—or written text to be more specific—this is not trivial to define due to the unlimited combinatorial nature of language. A natural way to overcome this is to consider sequences of a maximum fixed length T and to factorise the joint probability distribution into conditionals defined over its sub-parts: words or tokens. Additionally, it is intuitive to model dependencies between these sub-parts in a left-to-right manner: the probability of a word $\mathbf{x}_i^{(i)}$ depends on the words that precede it (i.e. its prefix $\mathbf{x}_{<j}^{(i)}$). The probability of a sequence is then defined as the joint probability of a sequence of words that factorises auto-regressively:

$$p(\mathbf{x}^{(i)}) = p(\{\mathbf{x}_1^{(i)}, ..., \mathbf{x}_T^{(i)}\}) = \prod_{j=1}^{T} p(\mathbf{x}_j^{(i)} | \mathbf{x}_{<j}^{(i)}) \tag{2.2}$$

We can further view the probability of a word as the probability under a categorical distribution with the size of our fixed vocabulary $|V|$. Thus, the final objective can be defined

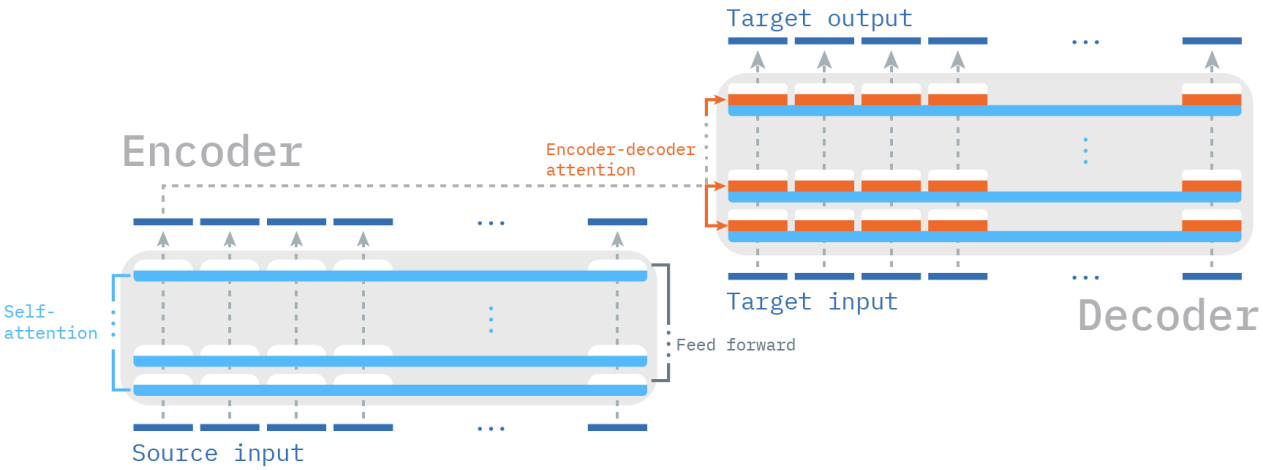

FIGURE 2.1: Overview of the architecture of the Transformer (Vaswani et al., 2017)

as finding the best setting of parameters θ to map a prefix $\mathbf{x}_{<j}^{(i)}$ of preceding words to the parameters of a categorical distribution over words for all words in the sequence, with some function f for which we consider (deep) neural networks:

$$\max_{\theta} \; p_\theta(\mathbf{X}) = \max_{\theta} \; \log \prod_{i=1}^{N} p_\theta(\mathbf{x}^{(i)}) = \max_{\theta} \; \sum_{i=1}^{N} \sum_{j=1}^{T} \log \mathrm{Cat}\left(\mathbf{x}_j^{(i)} | f(\mathbf{x}_{<j}^{(i)}, \theta)\right) \tag{2.3}$$

2.1.1 The Transformer

As this thesis considers a class of VAEs with a Transformer as architectural backbone, or as instantiation of f in Equation 2.3, we will now give a short review on it. The initial architecture as proposed by Vaswani et al. (2017) consists of an encoder and decoder and is designed for sequence-to-sequence modelling tasks such as neural machine translation. The model, in contrast with recurrent architectures, processes its inputs in parallel during training, making computation more efficient given the powerful parallel processing units that are available today. Interdependence between inputs is modelled by a so-called self-attention mechanism and positional embeddings rather than by recurrence. The model's encoder and decoder are both composed of multiple layers, all consisting of the same subcomponents (for an overview, see Figure 2.1). The most prominent subcomponents that live in both the encoder and decoder are the self-attention and feed-forward blocks. The decoder carries an additional encoder-decoder attention block in between those two. The architecture consists of other important elements as well, such as positional embeddings and layer normalisation units, but for details on those we refer the reader to the original work of Vaswani et al. (2017) as its not directly relevant to the subject of this thesis.

We will proceed to explain the self-attention mechanism, which is essential for understanding the TransformerVAE as implemented by C. Li et al. (2020). Because all inputs in a Transformer are processed in parallel, every layer receives hidden states corresponding to token positions and passes them on to the next. The number of inputs and hidden states flowing through the model is thus constant and fixed. A self-attention block (mathematically expressed in Equation 2.4) initially transforms an incoming hidden representation into a key-, a value- and a query vector. The weights of this transformation are learned. A query

vector belonging to a specific position is matched against key vectors of all positions in the sequence by means of a scaled dot product (including to its own key vector). This results in a scalar that expresses "compatibility" of other hidden states with the hidden state at issue. All these scalars that are associated with token positions are projected to the simplex by applying a SoftMax operation and then used for a convex combination of hidden states that results in a new "contextualised" hidden state. This mechanism does not act on the whole hidden state that is the input to that layer, but happens N times (often referred to as N *attention heads*): on N subvectors of the hidden state. By splitting the mechanism up, the meaning of "compatibility" may be interpreted and executed differently in each head, allowing for specialisation on specific information extraction and combination in specific heads, for example positional, syntactical or rare word attention as identified by Voita, Talbot, Moiseev, Sennrich, and Titov (2019).

$$\text{Self-attention}(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.4}$$

Encoder-decoder attention is the exact same mechanism, but forms query vectors from hidden states in the decoder, while the key-, value vectors are formed from hidden states in the encoder. This connects the encoder and the decoder and allows information to flow from the encoder to the decoder. Just like the encoder, the decoder receives input in the form of embeddings, but these inputs come from the target sequence. In machine translation this would for example be the target language to which one wants to translate to. To make sure the decoder is only to include target context from the past (left of the current prediction position) in its self-attention mechanism, the attention weights to future tokens are masked. It is important to note that, for MLE training, the target input at the decoder consists of ground-truth information during training (also known as *teacher-forcing*), allowing for parallel processing of the inputs. At prediction time this should be changed to conditioning on its own predictions. But, since conditioning predictions on its previous predictions is inherently sequential, processing at prediction time is recurrent as well.

Although the transformer model was originally designed for sequence-to-sequence tasks, comprising of an encoder and a decoder, an encoder-only adaptation (Devlin et al., 2019) and decoder-only adaptation (Peters et al., 2018) with unsupervised, or self-supervised objectives, were soon to follow. Both of these works rely mostly on the same building blocks as the original transformer by Vaswani et al. (2017). The former effort, called BERT, adopted a masked language modelling as auxiliary pre-training task to learn bi-directional token and sequence representations, that showed to be effective for down-stream tasks such as question answering and textual entailment. The latter, called GPT, used an unsupervised language modelling objective (next token prediction) and later (and much larger) versions have shown some impressive results, even in few-shot situations without any additional 'learning' (Brown et al., 2020). Y. Liu et al. (2019) made an effort, that goes by the name RoBERTa (used in this thesis), to improve and prolong the training procedure of BERT and showed its underestimated potential. Many follow-up works have been published since. Some of those works altered the architecture, for example the work by Dai et al. (2020) marrying the transformer and recurrence for processing longer and variable length sequences and the work by Child, Gray, Radford, and Sutskever (2019) sparsifying the attention mechanism with the goal of increasing its efficiency. Another line of work focused on improving the

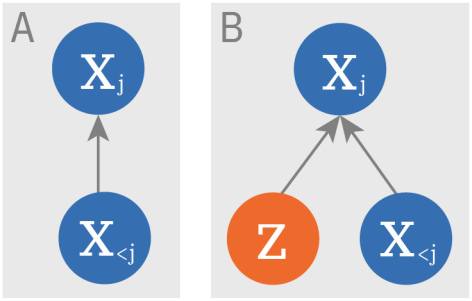

FIGURE 2.2: Changing the graphical model from a regular auto-regressive language model (A) to the graphical model of a latent variable auto-regressive language model (B)

(pre-)training objectives, such as Yang et al. (2019) who generalised auto-regressive modelling (factorising the joint distributions of a sequence in a directed way) from left-to-right to arbitrary uni-directional orders. Others focused on using existing pre-trained models for transfer learning applications. Rothe et al. (2020) showed how combinations of existing pre-trained models transferred well for a diverse set of sequence-to sequence tasks. Another work by Q. Liu et al. (2020) showed successful transfer learning to the dialogue modelling domain.

2.2 Deep generative latent variable language models

In Section 2.1 we defined a probabilistic language model in which we can employ a (deep) neural network to map a language prefix to the parameters of a categorical distribution over words in our vocabulary. We arrived at a statistical objective we can aim to optimise with respect to the parameters of our model (Equation 2.3). For reasons explained in the Introduction (Chapter 1), we set out to incorporate latent variables in the graphical model that constitutes this probability. That is, to change our graphical model from the one visualised in Figure 2.2 (A) into the one visualised in Figure 2.2 (B). The probability of a datum under a model with parameters θ can now be defined as the marginalisation of the latent variables out of the joint distribution between observed and latent variables. The probability the model assigns to a datum can thus be expressed as:

$$p_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int_{\mathbf{z}} p(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z} = \int_{\mathbf{z}} p(\mathbf{z}) \prod_{j=1}^{T} \text{Cat}\left(\mathbf{x}_j | f(\mathbf{x}_{<j}, \mathbf{z}, \theta)\right) d\mathbf{z} \tag{2.5}$$

As the integral, rather than summation, in Equation 2.5 already reveals, we consider these latent variables to be continuous. This is a logical choice because we are interested in dimensionality reduction, rather than clustering or revealing explicit discrete structure in the data, which is typical for discrete and structured latent variables respectively (Kim et al., 2018). A famous example of a discrete latent variable model in the context of language is Latent Dirichlet Allocation, which is a so-called *topic model* that defines a distribution over words given some cluster assignment specified by the latent variable (Blei, Ng, & Jordan, 2003). More recently, researchers are also trying to understand how deep neural networks can learn explicit discrete structures that are implicit in textual data, such as syntactic parse trees (Mihaylova, Niculae, & Martins, 2020).

Evaluating 2.5 is typically intractable, which means we cannot perform exact posterior inference learning algorithms such as Expectation Maximisation (Hartley, 1958; Dempster, Laird, & Rubin, 1977). For that reason, we turn to variational inference, a technique to perform approximate posterior inference (Jordan & Saul, 1999). Specifically, we will use a Variational Autoencoder, where the tractable surrogate of the posterior is modelled with a deep neural network. This framework will be covered in the next section in more detail.

2.2.1 The Variational Autoencoder

As introduced on a high level in the introduction (Chapter 1), VAEs are latent variable models that marry variational inference with deep neural networks. The networks are used to model complex conditional probability distributions and can be optimised with stochastic gradient descent (Kingma & Welling, 2019). Variational inference can be used in situations where integrals form an obstacle because they are intractable to compute or hard to differentiate, as we saw is the case when evaluating the probability defined in Equation 2.5. Let us simplify the notation of this intractable objective, abstracting away from the fact we are dealing with sequences and express it in expectation over the data. The objective that maximises the expected log probability of the data with respect to the model parameters θ can then be written as:

$$\max_{\theta} \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] = \max_{\theta} \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}\right] \tag{2.6}$$

To get around the intractability issue, we introduce an approximation to the true posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$: $q_\phi(\mathbf{z}|\mathbf{x})$ which holds its own parameters ϕ. The model further assumes the empirical availability of the (implicit) data distribution $p_{\mathcal{D}}(\mathbf{x})$, i.e. a dataset, and the availability of a prior distribution over the latent space $p(\mathbf{z})$. We further assume our approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ to be a factorised Gaussian (i.e. having independent dimensions and thus a diagonal covariance matrix). The prior is set to be a standard isotropic Gaussian: $\mathcal{N}(\mathbf{0}, \mathbf{I})$. With this set-up, the joint between the observed and latent variables is conveniently factorised in two ways, in the encoding or inference direction with the approximate posterior $q_\phi(\mathbf{x}, \mathbf{z}) = p_{\mathcal{D}}(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})$, which allows us to sample $\mathbf{z}$, and in the decoding or generative direction $p_\theta(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$, which allows us to sample $\mathbf{x}$. This type of inference, where we use a parameterised function with one set of parameters to predict the parameters of the variational distributions for all data points is called *amortised inference*. The cost of inference is shared, or amortised, across data points. Refer to Figure 1.1 in the Introduction for a visualisation of this two-way view of the joint distribution of $\mathbf{x}$ and $\mathbf{z}$.

Apart from the convenience of being able to efficiently sample $\mathbf{x}$ and $\mathbf{z}$, the approximate posterior turns an intractable objective (Equation 2.6) into an optimisation of a variational lower bound on the log probability the model assigns to the data (called the *evidence lower bound*, or ELBO). This is equivalent to the minimisation of the KL-divergence, or relative entropy, from $p_\theta(\mathbf{x}, \mathbf{z})$ to $q_\phi(\mathbf{x}, \mathbf{z})$. A KL-divergence expresses the closeness of two distributions as an expectation in information difference. Note that this quantity is non-negative, exactly zero if the two distributions are the same and non-symmetric (hence the name *divergence* rather than *metric*). This minimisation enforces consistency between the inference direction and generative direction. The equivalence (in terms of optimisation) between balancing the joints and maximising ELBO is shown below and follows the derivation from S. Zhao et al.

(2018):

$$\min_{\phi,\theta} D_{KL}\left(q_\phi(\mathbf{x},\mathbf{z})||p_\theta(\mathbf{x},\mathbf{z})\right) \tag{2.7}$$

$$= \max_{\phi,\theta} -D_{KL}\left(q_\phi(\mathbf{x},\mathbf{z})||p_\theta(\mathbf{x},\mathbf{z})\right) \tag{2.8}$$

$$= \max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{z}|\mathbf{x}) p_{\mathcal{D}}(\mathbf{x}) - \log p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z})\right] \tag{2.9}$$

$$= \max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right] + \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\log p_{\mathcal{D}}(\mathbf{x})\right] \tag{2.10}$$

$$= \max_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right] + \text{ constant} \tag{2.11}$$

$$\max_{\phi,\theta} \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\mathcal{L}^{\text{ELBO}}_{\mathbf{x}}(\phi,\theta)\right], \text{ where } \mathcal{L}^{\text{ELBO}}_{\mathbf{x}}(\phi,\theta) \leq \log p_\theta(\mathbf{x}) \tag{2.12}$$

The inequality (Equation 2.12) is due to the fact that the gap between the ELBO and the log probability of the data under the model is quantified as the (non-negative) KL-divergence from the true posterior to the approximate posterior. And therefore it holds that:

$$\log p_\theta(\mathbf{x}) = \mathcal{L}^{\text{ELBO}}_{\mathbf{x}}(\phi,\theta) + D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) \tag{2.13}$$

$$\text{and } D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) \geq 0 \tag{2.14}$$

$$\text{makes that } \log p_\theta(\mathbf{x}) \geq \mathcal{L}^{\text{ELBO}}_{\mathbf{x}}(\phi,\theta) \tag{2.15}$$

So, the KL-divergence from the true posterior to the approximate posterior embodies both the quality of the inference model and the gap between the lower bound and the intractable likelihood, which is often called the *tightness* of the bound (Kingma & Welling, 2019). This means that when the inference model is off, the bound is loose and optimisation of the lower bound might not have the intended effect of pushing up the probability the model assigns to the data. For a derivation that shows the validity of the addition in Equation 2.13 we refer the reader to Section 2.2. of the work by Kingma and Welling (2019).

Another useful way of writing the ELBO is as follows:

$$\mathcal{L}^{\text{ELBO}}_{\mathbf{x}}(\phi,\theta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right) \tag{2.16}$$

$$\mathcal{L}^{\text{ELBO}}_{\mathbf{x}}(\phi,\theta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z}) - \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}\right] \tag{2.17}$$

As can be seen in Equation 2.16, the ELBO consists of two terms. The first term is the cross-entropy of the generative distribution relative to the approximate inference distribution. This term measures the likelihood of $\mathbf{z}$ under the inference model and can be interpreted as how well the model reconstructs samples that are given to the model, hence its alias in negated form: the *reconstruction loss*. The second term is the KL-divergence from the prior to the approximate posterior.

Reparameterisation trick

For the ELBO to be optimised with stochastic gradient descent, one must be able to calculate the gradients of the loss with respect to the parameters of the model. In a VAE the parameters of the encoder ϕ are jointly optimised with the parameters of θ. As can be seen from

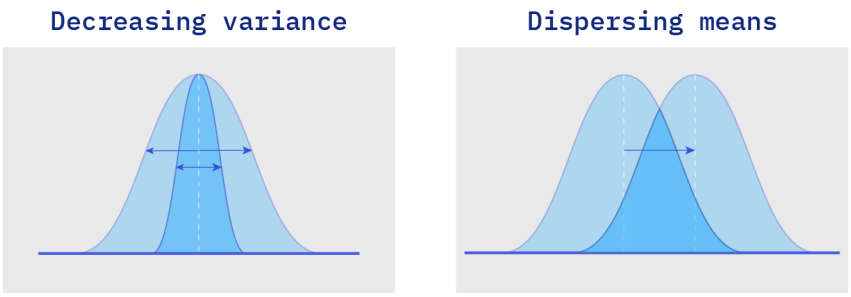

FIGURE 2.3: Two changes in Gaussian approximate posterior distributions that lead to distinguishability between encodings

the Equation 2.17, the ELBO is defined in terms of an expectation over the approximate posterior. This is not problematic for the parameters θ as this expectation does not depend on those parameters, but it is for the parameters ϕ. In the case of a Gaussian approximate posterior, this gradient of the expectation can be avoided by transforming the expectation into an expectation of an external noise variable, which allows us to use a Monte Carlo estimate. This change of variable, called the *reparameterisation trick* (Kingma & Welling, 2014; Rezende, Mohamed, & Wierstra, 2014; Titsias & Lázaro-Gredilla, 2014), is essentially changing sampling the latent variable into a transformation of the external noise variable through which we can back-propagate:

$$\text{Sampling: } \mathbf{z} \sim p(\mathbf{z}|\mathbf{x}) \tag{2.18}$$

$$\downarrow$$

$$\text{Reparameterisation trick: } \boldsymbol{\epsilon} \sim N(0, \boldsymbol{I}) \rightarrow \mathbf{z} = \boldsymbol{\epsilon}\sigma + \boldsymbol{\mu} \tag{2.19}$$

Modulating the bottleneck capacity with β-VAE

A popular modification of the ELBO inserts an additional hyperparameter β to the objective, which yield the $\mathcal{L}_{\beta-VAE}$ objective as first introduced by Higgins et al. (2016):

$$\mathcal{L}_{\beta-VAE} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \beta D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right) \tag{2.20}$$

For the generative model to distinguish between latent representations belonging to different data points (and thus to receive information from them), the inference model needs to make the posterior distributions different from each other and as a consequence relative to the prior space. This can be done by increasing the spread of the posterior means over different data points or by decreasing the variance of individual posterior means (Burgess et al., 2018). Both of these actions (visualised in Figure 2.3) decrease the overlap between posteriors belonging to different data points and thus increase distinguishability between encodings. They also increase the KL-divergence from the prior to the approximate posterior (the second term of the ELBO), which can thus be interpreted as the cost of encoding. And, by varying β one essentially varies the weight of the encoding cost, or the *capacity* of the information bottleneck (Burgess et al., 2018).

This objective has been introduced with the goal of learning *disentangled* representations, where the commonly conceived goal is to learn representations with independent feature dimensions (Mathieu, Rainforth, Siddharth, & Teh, 2019). The idea is that if β is set to values $\gg 1$, the axes of the learned representations become aligned with the generative factors of

variation of the data (Higgins et al., 2016). Burgess et al. (2018) explain this effect by reasoning that, primarily, the increased encoding cost forces the model to efficiently encode features that have the largest effect on the reconstruction term $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]$. And, these features intuitively correspond to more abstract features with a high global impact. Seen from this point of view, we get a more concrete idea of what the somewhat vague concept of generative factors of variation might entail. Secondly, they attribute features aligning with the axes to the fact that we encourage the posterior to be a factorised distribution. A way to efficiently encode very different generative factors with independent dimensions, is to split them up over different dimensions. S. Zhao et al. (2018) show with a Lagrangian perspective on latent variable models, that setting $\beta > 1$ yield a minimisation of the mutual information between $\mathbf{x}$ and $\mathbf{z}$, which is consistent with the idea of increased encoding cost and disentanglement learning. When the cost is increased, more impactful (in terms of reconstruction loss) features are encouraged to be encoded in an independent way. Although this idea of increasing the force of the bottleneck to encode more abstract and fundamental features of the data is conceptually appealing, it can be harmful in settings with auto-regressive decoders that can learn to model the data without making use of the latent variable.

The rate-distortion plane

When taking an information theoretic perspective on the ELBO, we can, in expectation over the data, recognise the information theoretic quantities distortion D and rate R in the reconstruction loss and KL-divergence from the prior to the approximate posterior respectively (Alemi et al., 2018a). They relate those quantities to the mutual information between $\mathbf{x}$ and $\mathbf{z}$ under q_ϕ, as their variational bounds, together with the fixed data entropy H:

$$D = -\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]\right] \tag{2.21}$$

$$R = -\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right] \tag{2.22}$$

$$H = -\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\log p_\mathcal{D}(\mathbf{x})\right] \tag{2.23}$$

$$H - D \leq I_q(X;Z) \leq R \tag{2.24}$$

From these bounds and the fact that $\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\mathcal{L}^{\text{ELBO}}_{\mathbf{x}}(\phi,\theta)\right] = -(R+D)$ draw some interesting conclusions. As we can see from the definition of the ELBO, the same ELBO value can be achieved with different rate distortion ratios. This means that ELBO optimisation itself is not sensitive to the relative size of these quantities. Additionally, they note the fact that rate upper bounds the mutual information and that the ratio between the two determines the range in which the mutual information can exist. They argue is thus important to target models with non-vanished rate to avoid negligible mutual information between $\mathbf{x}$ and $\mathbf{z}$ under q_ϕ and to report on rate and distortion separately, as the ELBO might be maximised with undesirable ratios between those two (read: with collapsed posteriors). We will revisit this theory in the next Section 2.3 when covering posterior collapse and also expand on it in our analysis in Chapter 4.

2.2.2 Useful decompositions of the rate term

The most common way of writing the ELBO is as the decomposition of the negative reconstruction loss and the KL-divergence from the prior to the approximate posterior (Equation 2.16). In this section we will highlight two further decompositions of the latter term that will

be referred to and reasoned about in our analysis in Chapter 4. To start with the decomposition of Hoffman and Johnson (2016). They split up the empirical notation of the KL term in a marginal KL term and an index-code mutual information term:

$$\frac{1}{N}\sum_{i=1}^{N} D_{KL}\left(q_\phi(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i)\right) = \underbrace{D_{KL}\left(q_\phi(\mathbf{z})||p(\mathbf{z})\right)}_{\text{marginal KL}} + \underbrace{\left(\log N - \mathbb{E}_{q_\phi(\mathbf{z})}\left[\mathbb{H}\left[q_\phi(\mathbf{x}|\mathbf{z})\right]\right]\right)}_{\text{Index-code mutual information}} \quad (2.25)$$

The *marginal KL* expresses the divergence from prior to the the marginal $q_\phi(\mathbf{z})$. The marginal $q_\phi(\mathbf{z}) = \int_{\mathbf{x}} q_\phi(\mathbf{x},\mathbf{z})d\mathbf{x}$ is the marginal induced by fixing the inference distribution q_ϕ. It can be seen as the average encoding distribution and is sometimes referred to as the *aggregated posterior* (Tomczak & Welling, 2017; Makhzani, Shlens, Jaitly, Goodfellow, & Frey, 2015). Conceptually, the marginal KL thus describes how close prior space is to the *average* encoding space. This is notably different from the KL-divergence we started at that measures the divergence from the prior to the *individual* encoding distributions. The second term is the index-code mutual information $I_q(X;Z)$ and quantifies the amount with which $\mathbf{z}$ varies with $\mathbf{x}$ under q_ϕ, or how much information can be known about $\mathbf{x}$ by observing $\mathbf{z}$. This decomposition highlights an important understanding with regards to trade-offs made internally in a VAE. In general we want the marginal KL to be low to meet our goal of learning a smooth latent space: on average we want our encodings to cover the whole prior space. The index-code mutual information term can be seen as a regulariser. This decomposition reveals that the ELBO objective aims to decrease this term and thus increase the overlap between encodings, which is conflicting with the reconstruction term in the ELBO in Equation 2.17.

A later work by T. Q. Chen, Li, Grosse, and Duvenaud (2018) decomposes the marginal KL term even further, assuming a factorised prior:

$$\frac{1}{N}\sum_{i=1}^{N} D_{KL}\left(q_\phi(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i)\right) = I_q(X;Z) + \underbrace{D_{KL}\left(q_\phi(\mathbf{z})||\prod_j q_\phi(\mathbf{z}_j)\right)}_{\text{total correlation}} + \underbrace{\sum_j D_{KL}\left(q(\mathbf{z}_j)||p(\mathbf{z}_j)\right)}_{\text{dimension-wise KL}}$$

The total correlation term quantifies the dependency between dimensions of the marginal $q_\phi(\mathbf{z})$ and can be seen as a generalisation of mutual information for a set of random variables $\mathbf{z}_j$ under q_ϕ. The dimension-wise KL term measures the divergence from the prior to the aggregated posterior per dimension. That is, reduced to a univariate evaluation. This decomposition was put forward from a disentanglement perspective. They argue that if the goal is to achieve independent dimensions, we might benefit focusing on the one term that is directly responsible for this: the total correlation term. For us it is useful to realise that if we use a factorised prior, the divergence of the encoding space from the prior space can be split up in a dimension wise divergence and a term that measures how entangled the dimensions of our average encoding space are.

2.3 Posterior collapse

Learning a VAE in practice can be hard due to the notorious issue that the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ learns to ignore the latent variable during optimisation. This event is often referred to as *posterior collapse*, *latent variable collapse* or—less aptly—*KL vanishing* and was first described by Bowman et al. (2016) in the context of language modelling. When the decoder has learnt

to ignore the latent variable, the decoder essentially degrades to a regular language model $p_\theta(\mathbf{x})$ that can deal with an extra noise input. In probabilistic terms, we saw in the Introduction (Chapter 1 and Equation 1.1) that the generative model degrades to a factorisation of marginals. And, since the likelihood of $\mathbf{z}$ is independent of the prior, reducing $p_\theta(\mathbf{x}|\mathbf{z})$ to $p_\theta(\mathbf{x})$, the *true* generative posterior consequently equals the prior, reducing $p_\theta(\mathbf{z}|\mathbf{x})$ to $p(\mathbf{z})$.

Importantly, we do not have a way of inspecting the true posterior and check whether it has collapsed to the prior. We can not inspect the true posterior due to intractability issues: this is the reason we turned to variational inference in the first place. What we can observe, however, is a symptom our model is likely to show. Recall that maximising the lower bound of the marginal log probability of the data under the model (i.e. the ELBO) is equivalent to minimising the gap between the approximate and true posterior (Equation 2.13). When the true posterior has collapsed to the prior, it is trivial for the approximate posterior to close this gap by mimicking the posterior in collapsing to the prior as well. This is something we can observe via the rate, or the expected KL divergence from the prior to the approximate posterior $\mathbb{E}_{p_\mathcal{D}(\mathbf{x})} D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)$. Thus, in practice, we diagnose posterior collapse by inspecting the rate: when it vanishes, we have reason to believe the true posterior has collapsed to the prior. Because the vanished rate is a consequence rather than the origin of posterior collapse, we think the term *KL vanishing* is less apt and should be avoided.

It now becomes clear that the extra noise input we mentioned previously, is in fact noisy because it looks like it comes from the prior, it carries no specific information about data points and is thus indistinguishable for the decoder. Since the approximate posterior is equal to the prior and the samples are indistinguishable, we can conclude $\mathbf{x}$ and $\mathbf{z}$ must be independent in the encoder as well. This is consistent with the fact that rate upper bounds the mutual information between the data and the latent in the encoder. If the rate vanishes, $\mathbf{x}$ and $\mathbf{z}$ must be independent:

$$I_q(X;Z) = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\Big[\log q_\phi(\mathbf{z}|\mathbf{x}) - \log q_\phi(\mathbf{z})\Big] = R = 0 \tag{2.26}$$

Additionally, since the gap between the approximate and true posterior is closed, our lower bound approaches the true log likelihood, which in turn is in line with the view that the decoder has reduced to a vanilla language model, which is an exact likelihood model.

2.3.1 Information preference property: the strong decoder problem

As we explained in the Introduction and as can be seen from the rate-distortion perspective on the ELBO, we know that ELBO optimisation itself is not sensitive to whether or not the posterior has collapsed. This can be observed even more directly from the marginal probability of the data under the model which is not a function of $\mathbf{z}$ to begin with. An important reason for the decoder not to use the latent representation is that decoders employed for sequences modelling are often strong decoders. Strong decoders in the context of language typically means they are auto-regressive models with a lot of capacity. In theory, any joint distribution over a sequence of text $p(\{\mathbf{x}_1, ..., \mathbf{x}_T\})$ can be modelled by a factorisation of conditionals $p(\mathbf{x}_i|\mathbf{x}_j)$ for any permutations of indices $\{1, ..., N\}$. In practice, the constraining factor for us to model such a factorisation is the parametric family we choose to model the conditionals $p(\mathbf{x}_i|\mathbf{x}_j)$ with. We also know that, for language, the categorical distribution is maximally expressive given the assumptions of a finite vocabulary and non-zero probabilities. So, an auto-regressive language model using the categorical distribution is theoretically

maximally expressive in modelling language. And thus, it is not unlikely for the decoder, given enough capacity, to model the data distribution well on its own, without making use of the latent variables. We remind the reader at this point that although this might seem a good achievement, this is not solely what we are after. We specifically aim at learning a data generative process that that explains the data well *and*, crucially, exploits meaningful patterns at a higher level, governed by our latent representations.

X. Chen et al. (2017) bring forward the bits back decoding argument to explain this phenomenon. They show that if the decoder can model the data distribution perfectly on its own, the global optimum is one that does not make use of the latent variables. They argue that due to the likely existence of a non-negligible posterior gap, the code coming from the encoder is an inefficient one. If the decoder can model dependencies on its own, it will prefer to do so instead of using the inefficient code coming from the encoder. This is why the *strong decoder problem* is sometimes referred to as the *information preference property* (S. Zhao et al., 2017) which highlights a slightly different, but important interpretation. In other words: the presence of a strong decoder facilitates the information preference property to take effect.

2.3.2 Methods to counteract posterior collapse

In this section, we will give a review of prevalent methods found in literature to counteract posterior collapse. It is important to keep in mind that it is impossible for us to act on the posterior directly, so techniques presented in this section aim to de-collapse the posterior in indirect ways. They do so in broadly five directions: 1) biasing away from zero-rate solutions 2) diminishing the symptom of a vanished rate or eliminating these solutions from the space of solutions, 3) increasing the efficiency of the latent code 4) decreasing the effect of the information preference property and 5) moving away from the (for representation learning) ill-positioned ELBO objective.

KL annealing

KL annealing was first introduced in the paper by Bowman et al. (2016) and entails annealing the relative weight of the KL term in the ELBO from zero to one over the course of training. This is equivalent to annealing β from β-VAE objective from zero to one. It is important to note that, when this weight is less than one, the ELBO is not guaranteed to be a valid lower bound anymore. By lowering the weight below one, the subtracted term in Equation 2.16 decreases and the bound increases. The intuition behind this strategy is that if the cost of encoding is less at the beginning of training, the encoder might have more incentive to encode information, instead of optimising directly to a zero rate solution directly. This technique is expanded upon in the work by Fu et al. (2019) to a cyclical schedule who argue that if annealing is iterated upon, the incentive to encode information is increased iteratively as well.

Weakening the decoder

As we have identified that a strong decoder can be a problem in modelling an interesting latent space, an obvious way to overcome this is to weaken the decoder. One simple interpretation of this in the context of language is to drop tokens from the prefix input at the decoder (Bowman et al., 2016). By doing this, the decoder has more incentive to seek for information in the latent representation as the contextual information is now incomplete. This

idea has been re-interpreted in the context of image modelling by X. Chen et al. (2017) in a more principled way: by limiting the kernel size in a convolutional neural network (CNN) the dependencies that may be modelled by the decoder are not weakened at random, but are reduced to spatially local dependencies. In this way, the information that can not be found locally, has to be encoded in the latent representation. Yang, Hu, Salakhutdinov, and Berg-Kirkpatrick (2017) transfer this idea to the language domain by implementing a language model with a CNN and using dilated convolutions. Further examples include the work by D. Liu and Liu (2019) that weaken a Transformer-based VAE by not incorporating any contextual information and the work on modelling music by Roberts, Engel, Raffel, Hawthorne, and Eck (2018) that weaken the decoder by limiting the scope of the decoder by means of a conductor RNN.

Richer priors

Another line of research argues we can try to use richer priors in the context of VAEs (Tomczak & Welling, 2017; Alemi et al., 2018a; X. Chen et al., 2017; Hoffman & Johnson, 2016; Razavi, Vinyals, Van Den Oord, & Poole, 2019). The argumentation on why we need to use richer prior varies from work to work and it should be pointed out that not all modelling settings are equal across those works. Hoffman and Johnson (2016), for example, reason from a setting with weak decoders. They show empirical results that indicate high mutual information as well as high marginal KL and put forward that richer priors might make it 'easier' for the model to reduce marginal KL. Another line of research explores richer priors in a strong decoder setting and applies a mismatch-by-design argumentation (Pelsmaeker & Aziz, 2019; Razavi et al., 2019). The argumentation is that if the prior is richer, for example multi-modal, it is not possible for a uni-modal approximate posterior to match it, which causes the rate to increase. Additionally, it is reasoned that for the encoding space to match the prior on average (low marginal KL), encodings need to vary with the inputs which in turn increases the mutual information (Pelsmaeker & Aziz, 2019). Yet another—but related—point of view, is put forward by for example Alemi et al. (2018a), X. Chen et al. (2017) and Tomczak and Welling (2017). Those works argue that the prior might be too simplistic to be efficiently used by the decoder. If the code in the latent space can not have a strong enough effect on the output distribution, due to being inefficient, this might cause the decoder to ignore the latent code altogether. Tomczak and Welling (2017) show modelling gains experimentally using a Variational Mixture of Posteriors as prior (they call it the VampPrior) in the image modelling domain. Pelsmaeker and Aziz (2019) experiment with using a richer prior in the language modelling domain, but do not find it advantageous empirically.

Targeting rate

Because a vanishing rate is the most obvious symptom of posterior collapse and because we know ELBO optimisation itself is not sensitive to trade-offs between rate and distortion, many solutions have been proposed to target a specific rate during optimisation. Alemi et al. (2018a) propose to convert ELBO optimisation to aim for a specific rate by maximising the objective $-(D + |\sigma - R|)$ with respect to the model parameters θ and ϕ, where σ is the target rate. Burgess et al. (2018) propose the same objective with additional weighting factor γ, but reason from a disentanglement perspective, not a posterior collapse perspective. Another related method is that of Free bits (Kingma et al., 2016) that interprets this idea as removing the incentive to drive the rate below a certain target rate defined as the budget of Free bits

(λ in the Equation 2.27 below):

$$\mathcal{L}_{\mathbf{x}}^{\text{FB}}(\phi,\theta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] + \max\left(\lambda, D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right) \tag{2.27}$$

Because the KL term in Equation 2.27 is capped when dropping below the threshold λ the lower bound may rise and risks not being a lower bound anymore. Moreover, the Free bits objective includes a maximum operator, the transition from being below the Free bits level and above is a sudden one. X. Chen et al. (2017) propose to solve this by dynamically adapting the weight of the KL term depending on the rate with a method aptly called Soft Free bits. Pelsmaeker and Aziz (2019) take this idea a step further and view the objective as the following constraint optimisation problem:

$$\max_{\theta,\phi} \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\mathcal{L}_{\mathbf{x}}^{\text{ELBO}}(\phi,\theta)\right] \text{ s.t. } \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right] \geq r \tag{2.28}$$

They show this can be optimised by optimising the Lagrangian (with Lagrangian multiplier u), a method they call Minimum Desired Rate (MDR). This solves the problem of introducing extra hyper parameters as is the case for Soft Free bits. This objective maximises the Lagrangian with respect to the parameters of the model and solves the dual problem with minimisation. The primal objective is:

$$\max_{\theta,\phi} \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\mathcal{L}_{\mathbf{x}}^{\text{ELBO}}(\phi,\theta)\right] - u\left(r - \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)]\right) \tag{2.29}$$

Change of objective

Some works propose to change the objective altogether, moving away from the vanilla ELBO objective, because of its lack in preference on the mutual information between data and latent representations. S. Zhao et al. (2017) make this preference explicit by adding a mutual information term to the objective and rewriting it in the following form:

$$\begin{aligned}\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\mathcal{L}_{\mathbf{x}}^{\text{INFO}}(\phi,\theta)\right] = \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \\ (1-\alpha)\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right] - (\alpha+\lambda-1)D_{KL}\left(q_\phi(\mathbf{z})||p(\mathbf{z})\right)\end{aligned} \tag{2.30}$$

The last term in this Equation 2.30 is the marginal KL. It contains $q_\phi(\mathbf{z})$, which is intractable to compute and computationally expensive to approximate reliably with a Monte Carlo estimate $\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[q_\phi(\mathbf{z}|\mathbf{x})]$. To get around this issue S. Zhao et al. (2017) propose to use the differentiable two sample test statistic Maximum Mean Discrepancy (MMD) (Gretton, Borgwardt, Rasch, Schölkopf, & Smola, 2012) between samples from the marginal $q_\phi(\mathbf{z})$ obtained via ancestral sampling and samples from the prior $p(\mathbf{z})$.

A follow-up work in this direction is the work by S. Zhao et al. (2018), who unify many previous efforts in deep generative latent variable modelling (including InfoVAE) in a Lagrangian framework. In this framework the primal problem is set to maximise or minimise the variational bounds on mutual information w.r.t. the generative and amortised inference distribution, subject to the constraints of accurately modelling the data distribution and enforcing consistency between the two views on the joint distribution $q_\phi(\mathbf{x},\mathbf{z})$ and $p_\theta(\mathbf{x},\mathbf{z})$. They show that the objectives the framework unifies can be optimised as a relaxed Lagrangian dual function, instead of with fixed parameters. We will return to these objectives in Chapter 4 as they are closely related to the subject matter presented there.

Chapter 3

Implementation of the TransformerVAE

In this chapter we will specify the implementational details of the TransformerVAE we use as a basis for all the experiments conducted for this thesis research. We will start by describing the architecture that is based on the work of C. Li et al. (2020) in Section 3.1. In Section 3.2, we will describe adaptations made to the model to decrease the memory consumption of the model, other training details and make a note on the datasets used. An overview of the adapted implementation is visualised in Figure 3.1.

3.1 The TransformerVAE and memory mechanism

In Chapter 2 we have given background information on the Transformer and how it functions as a language model as well as an overview of what a VAE looks like. What is left for us to specify is how to use the Transformer architecture as a backbone for a VAE. As we have explained in Section 2.1.1, the original Transformer architecture by Vaswani et al. (2017) employs encoder-decoder attention, or cross-attention, to connect the encoder with the decoder. For the Transformer to function in a VAE setting, we need to replace this mechanism with a latent space, or a stochastic information bottleneck. This requires us to specify two things: 1) we need to specify a way to aggregate the input information flowing through the encoder and map it to the parameters of our approximate posterior distribution and 2) we need to specify a mechanism for the decoder to incorporate information from the sampled latent representation in its prediction process.

For the first requirement the TransformerVAE uses a commonly adopted method of prepending a special start token token and appending a special end-of-sequence token to the input sequence and treating a concatenation of the last hidden states corresponding to these token positions as a an aggregated sequence representation. This idea has first been proposed by Devlin et al. (2019) that made an encoder-only adaptation of the Transformer designed for representation learning that serves language understanding tasks. To map this concatenation of hidden states to the parameters of our approximate posterior distribution, we add an additional dense pooling layer.

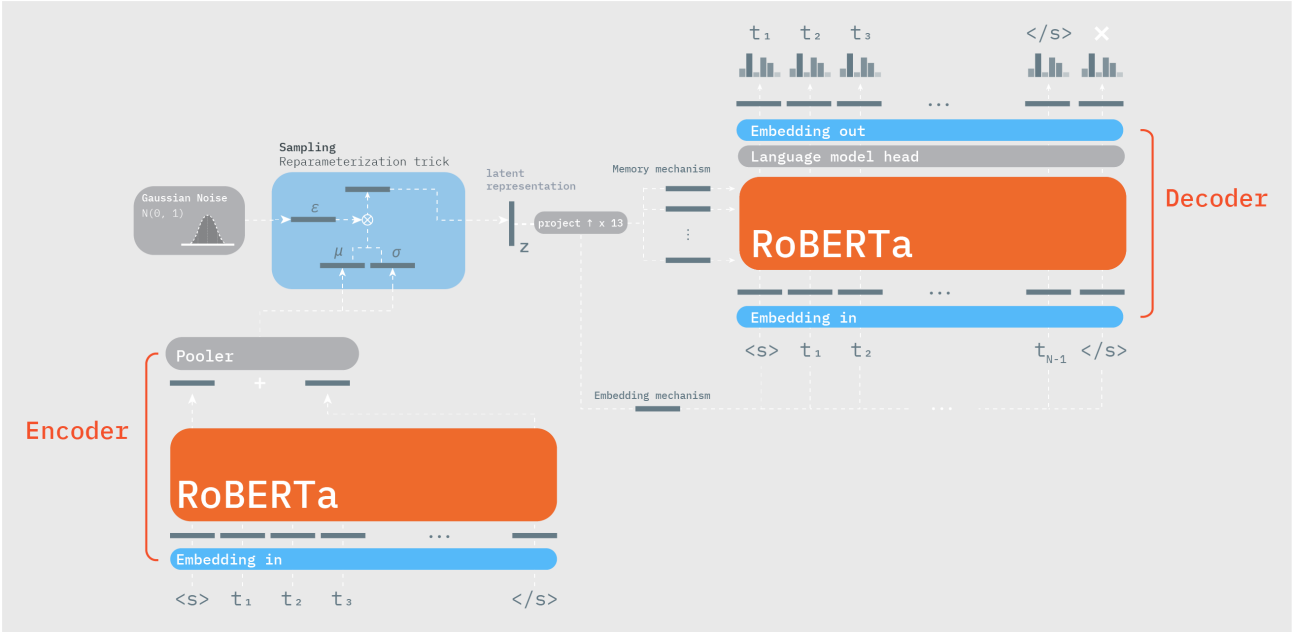

FIGURE 3.1: The overview of the model used in this thesis: an adapted version of the model by C. Li et al. (2020)

For the second requirement C. Li et al. (2020) propose two mechanisms: the *embedding mechanism* and the *memory mechanism*. The former mechanism simply entails adding the latent representation to the input embeddings of the contextual information presented to the decoder. This requires us to project the latent representation (32) to the correct dimensionality of the input embeddings (768). The latter mechanism incorporates the latent representation in the decoder via the self-attention mechanism at every layer. For this mechanism, the latent representation also needs to be projected to match shapes: to the dimensionality of the hidden states times the number of layers ($32 \rightarrow 768 \times 12$). The memory mechanism then adds the latent representation to self-attention mechanism by injecting it in the form of key and of value vector, but not as query vector. This makes that the hidden states in the decoder are contextualised with information from the encoder. In all experiments conducted in this thesis, both mechanisms are considered active by default.

3.2 Architectural adaptations and training details

C. Li et al. (2020) use a trained BERT model to initialise their encoder (Devlin et al., 2019) and a trained GPT-2 model to initialise their decoder (Wu & Lode, 2020), which together yields a model consisting of approximately 230M parameters. This is computationally very demanding, so we propose to initialise both the encoder and decoder with the same checkpoint and tie the weights between them. We initialise both with a RoBERTa checkpoint (Y. Liu et al., 2019)[1]. This cuts the number of parameters back to 125M and comes with the additional advantage that both encoder and decoder share the same tokenisation scheme: the Byte-Pair-Encoding scheme (Sennrich, Haddow, & Birch, 2016). Additionally, since our encoder and decoder share the same embedding matrices, all embedding matrices can be tied as well: the input embedding matrix of the encoder, the input embedding matrix of the decoder and

[1] We use implementations and hosted checkpoints by Hugging Face, specifically we use the checkpoint `roberta-base` with a hidden state dimensionality of 768 and 12 layers

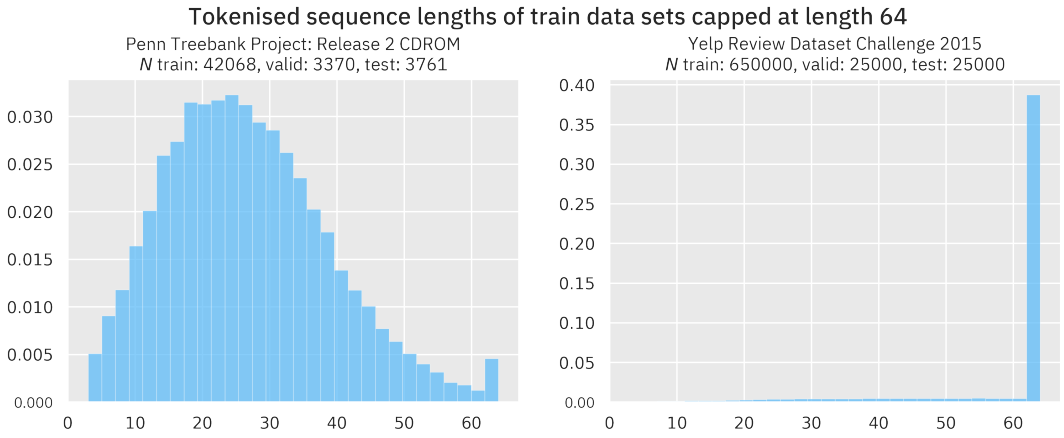

FIGURE 3.2: Length distributions of datasets used

the output embedding matrix of the decoder. The embedding matrices are *not* fixed during training.

Most of the memory gains result from tying the weights of the encoder and decoder and the embedding matrices, but the model still demands a lot of memory to train. To reduce the memory consumption and increase the effective batch size we:

- Employ *mixed precision training*, which means that some matrix operations are implemented with half precision rather than single precision. To safely implement this and make sure our gradients are scaled to avoid underflow we use the PyTorch implementation[2].
- Implement a training procedure that is *distributed across multiple GPUs*. Specifically the optimisation process is parallelised across multiple GPUs by running the exact same procedure on the different GPUs and share only the gradient data across them to simulate a training process powered by those GPUs combined. Or in other words, to effectively increase the batch size by the number of GPUs used. We work with PyTorch classes that are designed to support such distributed implementation[3]. We have access to 4 GPUs with 11GB memory each.
- *Accumulate gradients* over multiple train steps. By only updating our optimisers once every N steps, gradients accumulate and increase the effective batch size of the backward pass.

For all of our experiments we use a latent space of dimensionality 32 as is the standard in related research. We consider a maximum tokenised sequence length of 64. For most experiments we use the English Penn Treebank (PTB) dataset (Marcus, Santorini, & Marcinkiewicz, 1993)[4], which one of the most commonly used datasets in related work. In a few experiments we use an extra dataset: an English Yelp Review dataset (Zhang, Zhao, & Lecun, 2015)[5]. If not explicitly mentioned otherwise, the PTB dataset is used. In Figure 3.2 a length distribution for tokenised sequences with a maximum length of 64 is plotted for both datasets.

[2] Pytorch Cuda Automatic Mixed Precision Package
[3] Pytorch Distributed Data Parallel
[4] Specifically we use the one hosted by Hugging face: `ptb_text_only`
[5] Specifically we use the one hosted by Hugging face: `yelp_review_full`

Optimisation	Target rate	Distortion	Rate	ELBO	IW LL	IW LL (per token)	PPL
AE	0.0	70.54	160.09	-230.63	-222.19	-9.76	17,370.89
VAE	0.0	94.34	0.03	-94.37	-94.09	-3.5	33.23
CYC-FB	0.12	89.52	8.8	-98.32	-94.16	-3.51	33.38
CYC-FB	0.25	87.88	12.69	-100.57	-95.23	-3.56	35.05
CYC-FB	0.5	82.85	24.82	-107.67	-100.54	-3.8	44.76
CYC-FB	0.75	83.01	27.65	-110.67	-103.12	-3.92	50.44
CYC-FB	1.0	78.96	37.64	-116.6	-108.91	-4.19	65.78

TABLE 3.1: Results of runs on PTB validation set (N = 3370) that form the basis of the main Chapter 4. The number of importance weighted samples used for the calculation of importance weighted log likelihood (IW LL) and perplexity (PPL) is 500. The Free bits target rate displayed here is defined per dimension. CYC-FB stands for the model being optimised with a combination of Free bits and cyclical annealing KL annealing.

3.3 Importance weighted log likelihood of the base implementation

We mentioned in the Introduction (Chapter 1) that there is no specific intention to try and improve on the results as presented by C. Li et al. (2020). But, as it is rather common to report on log likelihood values and to show our implementation performs in a similar range (although slightly worse) as the values reported by C. Li et al. (2020), we report on those and other global statistics in in Table 3.1 for the PTB validation set. These runs form the base of most of the results presented in the next chapter. Additional results with different settings (e.g. with drop-out or the use of *only* the memory mechanism) and for both the datasets can be found in Appendix A.

Recall that, since our model is not an exact likelihood model, we need to turn to estimation. To estimate log probability the model assigns to the data $\log p_\theta(\mathbf{x})$, we can in theory use Monte Carlo sampling, but this would take an impractically large number of samples to cover a substantial amount of the latent space due to the curse of dimensionality. In order to improve on sampling complexity we can turn to importance sampling (where we set $N = 500$) (Burda, Grosse, & Salakhutdinov, 2016):

$$\log p_\theta(\mathbf{x}) = \int_z p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \approx \frac{1}{N}\sum_i^N \frac{p_\theta(\mathbf{x}|\mathbf{z}_i)p(\mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathbf{x})} \tag{3.1}$$

The final calculation of perplexity (PPL) is the exponent of the importance weighted negative log likelihood, averaged per token:

$$\text{PPL} = \exp\left(\frac{\sum_j^M -\log p_\theta(\mathbf{x}_j)}{\sum_j^M |\mathbf{x}_j|}\right) \tag{3.2}$$

Chapter 4

Assessing the TransformerVAE as a latent variable model

In this chapter, the main analysis of this thesis research is presented which follows the structure as laid out in the Introduction (Chapter 1). In the first Section 4.1 we will focus on the manifestation of the strong decoder problem in the TransformerVAE. In Section 4.2, we will explore an information theoretic view of VAEs and argue that there is an important axis not accounted for in most existing views that is directly relevant to the goal of this thesis: marginal KL. We will analyse existing methods along this axis in Section 4.2.1 and cover a conceptual view of the role this quantity might play during optimisation in Section 4.2.2. Lastly, in Section 4.3, we will investigate what kind of information can be encoded by the TransformerVAE and speculate on inductive biases of the memory mechanism (C. Li et al., 2020).

4.1 Demonstrating the information preference property: a TransformerVAE is a *very* strong decoder

In the Introduction (Chapter 1) we motivated that demonstrating the strong decoder effect in the TransformerVAE is useful. We argued that because C. Li et al. (2020) are not explicit on the indispensable role of strong decoders in causing posterior collapse, this might distract from the very goal of this line of research: using powerful density estimators in a VAE setting and thus *overcoming* the strong decoder problem. We know that large, pre-trained Transformer models are excellent at modelling data distributions of language and thus form an exemplar of the strong decoder class. And, even though the fact that the network is initialised with pre-trained checkpoints does not necessarily contribute to the strength of the decoder in a theoretical sense, in practice it might introduce a modelling bias that makes the strong decoder effect even more likely to occur. In other words, the weight initialisation does not make for the architecture to be able to model *different* dependencies within the data than with random initialisation, but it is definitely more likely to find them sooner.

Perhaps the most direct way of demonstrating the strength of the decoder is by weakening it and observing the effects on the information flow by inspecting the rate. Weakening the decoder was first explored by Bowman et al. (2016) in the form of word drop-out from the prefix presented to the decoder. In Section 2.3.2 we reviewed other, more principled

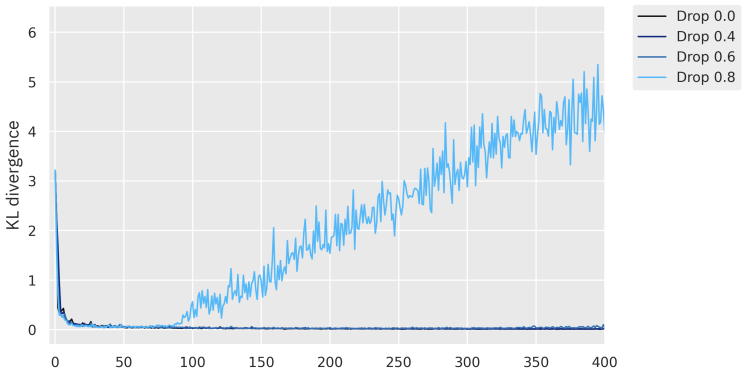

FIGURE 4.1: The rate plotted during ELBO optimisation for the Transformer-VAE with different levels of token drop-out at the decoder.

ways tried in literature to weaken the decoder. For our demonstrative purpose, however, a simple method like word drop-out is a suitable choice. It is perhaps an even more suitable method in the context of the Transformer architecture than in the context of recurrent architectures it was first tried in by Bowman et al. (2016). Because Transformers can process information in parallel, dropping tokens can be done gradually, while with recurrent architectures dropping an input at a time step is quite a strong intervention.

Figure 4.1 shows a plot of the KL-divergence from the prior to the approximate posterior in expectation over the data (i.e. the rate) during training for different levels of drop-out. The models are all trained with a vanilla VAE objective (i.e. maximisation of the ELBO). We can observe that the only model that starts using rate to transmit information from the encoder to the decoder is the model where 80% of the context information at the decoder is removed. Apparently, with lower levels of drop-out, the decoder can still find enough dependencies to explore locally and prefers to do that over making use of the information coming from the encoder.

4.1.1 Connecting optimisation techniques from C. Li et al. (2020) to the strong decoder problem

In their paper, C. Li et al. (2020) report on a combination of three techniques to combat posterior collapse: 1) the use of Free bits, 2) the use of cyclical KL annealing and 3) pre-training the latent space. The first two techniques we reviewed in Section 2.3.2. Pre-training a latent space is presented as a novel technique by C. Li et al. (2020). They claim that when pre-training a latent space on a large text corpus, posterior collapse can be reduced. It is important to note that pre-training and fine-tuning in this context entails training a VAE and is completely separate from initialisation with pre-trained language model checkpoints. Another important side note is that in both pre-train and fine-tune phase they employ the other techniques (Free bits and cyclical KL annealing) as well. We will now continue to assess the relative contribution of those techniques to bias away from zero rate solutions and connect it to the intuition we just built on role of the strong decoder. Our main hypothesis regarding those techniques is that Free bits is the indispensable factor in achieving substantial rate and that this can be understood from the information preference perspective.

KL annealing and latent space pre-training are both motivated from the point of view that if the information coming from the decoder at the beginning of training is of low quality, the decoder is not incentivised to use it. With KL annealing, we effectively decrease

	ELBO	Distortion	Rate
Cyclical annealing + Free bits (0.5 nats per dimension)	-192.39	168.01	24.38
Cyclical annealing	-181.88	179.91	1.97
Free bits (0.5 nats per dimension)	-183.27	163.80	19.47

TABLE 4.1: Ablation of the use of cyclical KL annealing versus the use of Free bits on the PTB validation set (N = 3370)

the cost of the rate at the beginning of training which potentially makes it easier to start transmitting valuable information to the decoder which may increase the incentive to use it. For latent space pre-training the idea is that the representations already contain good quality information from the pre-training phase, also making it more likely for the decoder to start using them. Free bits, on the other hand, changes the cost balance in the model in its entirety, effectively diminishing the cause of the information preference problem. The information preference problem dictates that if there is information being transmitted from the encoder to the decoder that can be modelled locally in the decoder, it will prefer to model it locally. This makes sense, as transmitting information comes at a cost expressed as rate. When we decrease this cost by granting the model a free budget, this information preference can be cancelled out. The model is thus more likely to use rate (up to the number of Free bits assigned) to transmit information because it comes at zero cost.

In Table 4.1 we report rate and distortion values of VAEs trained with making use of cyclical annealing, of Free bits and of both. We implemented cyclical annealing and Free bits closely following C. Li et al. (2020). Cyclical annealing is interpreted as splitting the epoch in three phases: the first half epoch the model is trained as an autoencoder ($\beta = 0$), the third quarter of the epoch the KL term is linearly annealed with a (β) weight increasing from 0 to 1. The last quarter of the epoch the model is optimised with a regular ELBO objective ($\beta = 1$). Concerning Free bits, we implement the dimension-wise version of Equation 2.27, where the KL term is thus implemented as written below. We will refer to the dimension wise Free bits level λ_d as the target rate (expressed in nats).

$$\sum_{i=1}^{D} \max(\lambda_d, D_{KL}\left(q(\mathbf{z}_i|\mathbf{x})||p(\mathbf{z}_i)\right)) \tag{4.1}$$

In Table 4.1, we can see a clear effect: when only cyclical annealing is used, the VAE stays in a low rate range. On the contrary, if Free bits are granted to the VAE the rate is increased substantially. Additionally, the table gives an indication of cyclical annealing further driving up the rate than with only using Free bits. We will return to this effect in experiments in later sections as this table can not reliably make definitive conclusions on this matter.

In Figure 4.2 we show the effect of pre-training a latent space. Here, we test the effect of pre-training a latent space by fine-tuning with a vanilla VAE objective. It is important to note that this is different from what the authors propose, but we think this is a fairer way to assess the potential of the technique. In our perspective, overcoming a vanished rate means that when applying the technique and no other technique its effect should hold. For this reason, we pre-train a latent space with cyclical annealing and Free bits and fine-tune without. If pre-training on its own was effective in overcoming posterior collapse it would mean that in fine-tuning phase, the rate would remain positive. We can see, however, that in the fine-tuning phase, the KL-divergence drops to zero again if no other tricks are applied. We argue that for both techniques, KL annealing and pre-training a latent space,

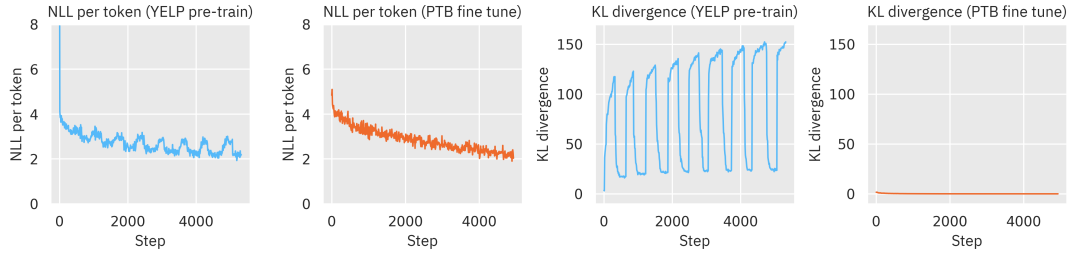

FIGURE 4.2: This figure shows the negative log likelihood (NLL) per token over the course of pre-training (blue) and fine-tuning (orange). We can observe KL-divergence drops to zero in the fine-tune phase.

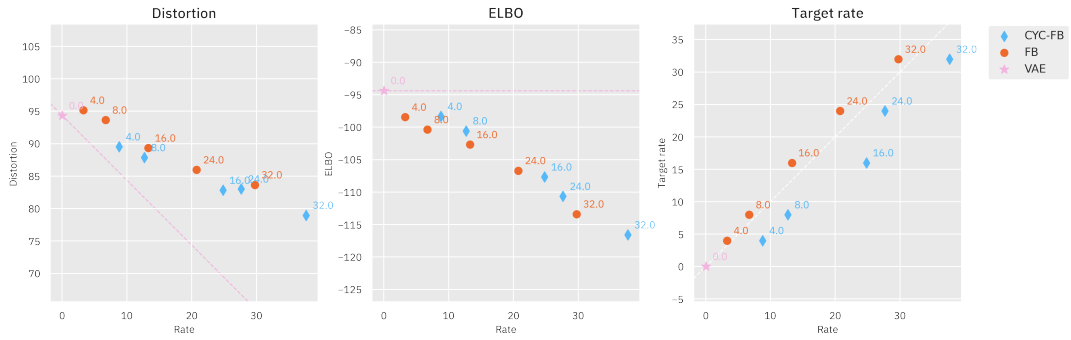

FIGURE 4.3: ELBO, Distortion and target rate plotted against Rate for different target rates (annotated in text) for the PTB validation set (N = 3370)

the information preference problem remains and both techniques are thus not able to combat posterior collapse on their own. It remains possible that the techniques further stimulate the information flow in the model, but we think it is fair to say they do not *solve* the problem of a collapsed posterior.

4.1.2 Strong decoders in the rate-distortion plane

Another way to investigate the nature of posterior collapse in the TransformerVAE is by inspecting the rate-distortion plane. We know that with a collapsed posterior, we are likely optimising towards a point on the axis where rate is zero. Additionally, we have observed that Free bits can lead us to points with non-zero rate. In Figure 4.3 we plot the rate against 1) the distortion, 2) the ELBO and 3) the target rate. The dashed pink lines indicate the fixed ELBO achieved with a collapsed VAE model. The first two plots tell the same story, one through distortion, the other through the ELBO. We can see in the ELBO plot that, with higher rates, our ELBO is worse. In the distortion plot, we can see that rate increases faster than the distortion decreases. This plot indicates that the model can lower distortion with rate and thus transmit information via the latent space. But, at the same time, this plot also indicates that the model has trouble using its encodings efficiently: distortion is not decreased in equal proportion to the increase of rate. This trend of the ELBO decreasing for higher rates is typical for strong decoder VAE models (Alemi et al., 2018a). The last plot in this Figure shows the relation between rate and target rate. We can again observe the effect that cyclical annealing drives the rate up above the target rate, while without it the rate stays a bit below the target rate. This suggests that cyclical annealing indeed biases towards exploration of *slightly* higher rate zones.

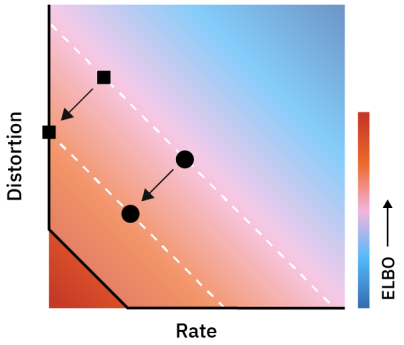

FIGURE 4.4: Two changes in the rate-distortion plane that are equivalent in terms of ELBO optimisation but differ crucially in rate levels.

4.1.3 Conclusion

So far, we have focused on inspecting rate as a symptom of a collapsed posterior and judged techniques to overcome posterior collapse in their ability to counteract this symptom in the TransformerVAE. We have observed that cyclical KL annealing and pre-training the latent space cannot achieve substantial rate on their own, while Free bits can. This can be understood from the information preference perspective, as Free bits changes the tipping point for this preference by changing the cost balance in the model. This does not directly imply, however, that Free bits optimisation has led to a "good" quality, uncollapsed VAE model: it just shows it has learnt a model that uses substantial rate. In other words, the experiments have shown it has achieved a substantial upper bound for the dependency between $\mathbf{x}$ and $\mathbf{z}$ tot exist in (goal 2 from the Introduction, Section 1.3), but the actual magnitude of the dependency (i.e. mutual information) remains unclear. Since we see the distortion decreasing for higher rates, we have reason to believe it is substantial, but we have also observed that the model is unable to effectively diminish the distortion with an amount that is proportional to this encoding cost, hinting at inefficient encodings. We will return to the mutual information quantity later in the next section (Section 4.2). Additionally—and perhaps more importantly—we will explore an additional axis of evaluation to give substance to what "good" means that goes beyond the VAE's potential to transmit information from the encoder to the decoder, concerning goal 4 from the Introduction: its ability to perform accurate approximate inference.

4.2 The missing axis in the information theoretic view of VAEs: the quality of approximate posterior inference

In the preceding chapters, we have established the fact that ELBO optimisation on its own does not encode our preferences with respect to representation learning. The ELBO is not sensitive to the existence of a dependency between $\mathbf{x}$ and $\mathbf{z}$. We can directly see that from Equation 2.6, which solely depends on $\mathbf{x}$. Another way of revealing this is via the rate-distortion view. Since the ELBO is not sensitive to the ratio between these two quantities, the ELBO may be maximised with low rate. Recall that rate upper bounds the mutual information $I_q(X;Z)$. So, a vanished rate means a vanished mutual information (dependency) between $\mathbf{x}$ and $\mathbf{z}$ under q_ϕ. This lack of dependency preference is visualised in Figure 4.4:

the two shifts of the square and circle visualised with arrows in the rate-distortion plane are equivalent in terms of ELBO optimisation, but the shift of the square results in a state with zero rate suggesting a collapsed posterior, while the other shift from the circle crucially does not.

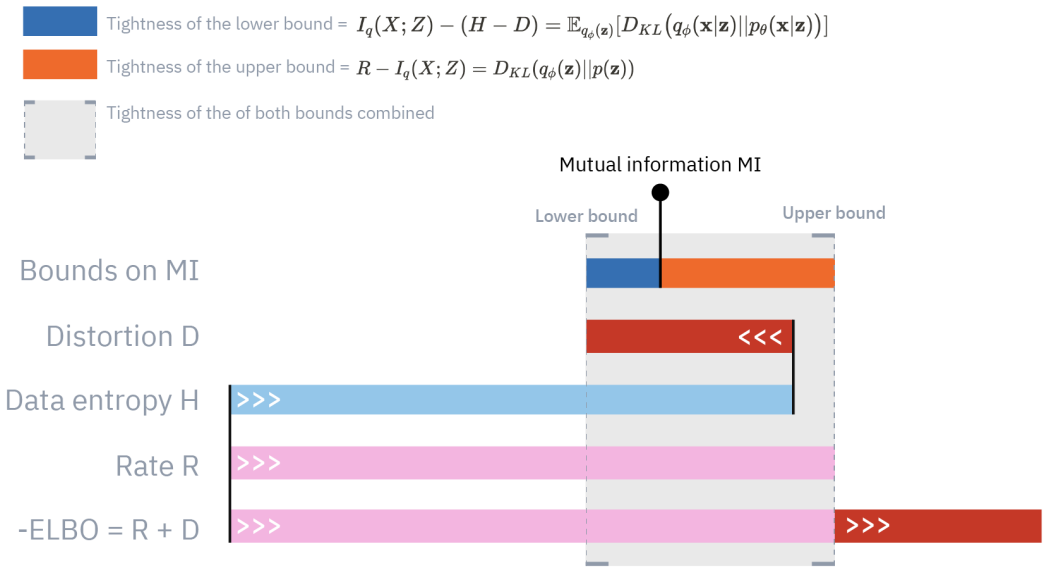

FIGURE 4.5: A visualisation of the variational bounds on mutual information as presented in Alemi et al. (2018a) and their relation to R, D and H.

We will proceed to further explore the relation between rate, distortion and mutual information and return to the variational bounds on mutual information as defined in Equation 2.24 in Section 2.2.1. In Figure 4.5 we visualise these relationships. We can see that the negative ELBO equals the sum of of rate R and distortion D. The data entropy H is a fixed quantity and beyond our control, but together with the distortion defines a lower bound on mutual information by the relation $I_q(X;Z) \geq H - D$ and therefore appears in the figure. This relation is also the reason the distortion is aligned on the right side with H and should be thought of as to increases to the left. The upper bound on mutual information is defined by the rate R. The space between the bounds is shaded light grey and the mutual information, the black line with circle on top, is by definition guaranteed to exists in between those bounds.

From the definition of the bounds, we can derive the tightness of the lower bound on mutual information as:

$$I_q(X;Z) - (H - D) = \tag{4.2}$$

$$\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z})p_\mathcal{D}(\mathbf{x})}\right] + \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\log p_\mathcal{D}(\mathbf{x})\right] - \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]\right] = \tag{4.3}$$

$$\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x}|\mathbf{z})q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})p_\mathcal{D}(\mathbf{x})}\right] + \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\log p_\mathcal{D}(\mathbf{x})\right] - \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \tag{4.4}$$

$$\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log q_\phi(\mathbf{x}|\mathbf{z}) - \log p_\theta(\mathbf{x}|\mathbf{z})\right] = \tag{4.5}$$

$$\mathbb{E}_{q_\phi(\mathbf{z})}\left[D_{KL}\left(q_\phi(\mathbf{x}|\mathbf{z})||p_\theta(\mathbf{x}|\mathbf{z})\right)\right] \tag{4.6}$$

This expected KL-divergence (Equation 4.6) is denoted with the dark blue bar in Figure 2.24. This quantity is a bit hard to grasp intuitively as it concerns the implicit distribution $q_\phi(\mathbf{x}|\mathbf{z})$, which is implicit due to the implicitness of $p_\mathcal{D}(\mathbf{x})$. It describes how well our decoder matches a hypothetical decoder that is induced from an inference point of view. What is important to realise about this quantity, is its relation to the distortion. If it is much smaller than the distortion, we know our decoder $p_\theta(\mathbf{x}|\mathbf{z})$ is overcompensating for an inefficient code that is sent by our encoder $q_\phi(\mathbf{z}|\mathbf{x})$. This is undesirable and a symptom of the strong decoder problem. So, even though it is a quantity that is low if the model is consistent $q_\phi(\mathbf{x}, \mathbf{z}) \approx p_\theta(\mathbf{x}, \mathbf{z})$, we do not want this quantity to be *unrealistically* low as it hints at the decoder overcompensating for the encoder.

Similarly, we can derive the tightness of the upper bound on mutual information as:

$$R - I_q(X; Z) = \tag{4.7}$$

$$\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\Big[D_{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\Big)\Big] - \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\left[\log \frac{q_\phi(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z})p_\mathcal{D}(\mathbf{x})}\right] = \tag{4.8}$$

$$\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\Big[\log q_\phi(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{z})\Big] - \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}\Big[\log q_\phi(\mathbf{z}|\mathbf{x}) - \log q_\phi(\mathbf{z})\Big] = \tag{4.9}$$

$$\mathbb{E}_{q_\phi(\mathbf{z})}\Big[\log q_\phi(\mathbf{z}) - \log p(\mathbf{z})\Big] = \tag{4.10}$$

$$D_{KL}\Big(q_\phi(\mathbf{z})||p(\mathbf{z})\Big) \tag{4.11}$$

We can recognise this KL-divergence (Equation 4.11 and denoted with the orange bar in Figure 4.5) as the marginal KL. Recall that it quantifies the divergence from the prior to the marginal $q_\phi(\mathbf{z})$, or average encoding distribution. This quantity, we argue, is of great interest for our latent variable model. For this quantity to be small, posterior inference must be accurate. That is to say that the encoder is optimal for the decoder we have, or the encoder predicts an approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ that approaches the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ induced by the generative model. It is important to note that this does not say anything about whether or not the decoder is exploiting any dependencies between observed and unobserved variables. But, it does say that if it *were* to exploit dependencies, the encoder will serve this mechanism efficiently. This is exactly what we are after: the encoder serving up encodings that are as efficient as possible from a decoder point of view.

By further inspecting the diagram displayed in Figure 2.24 we can note a few more things. First, the sum of the tightness of the lower bound to the mutual information plus the tightness of the upper bound to the mutual information, or the *combined tightness*, which is visualised as the light grey area, is uniquely defined by the rate and distortion in terms of magnitude *and* position. A fixed ELBO (itself insensitive to the rate-distortion ratio) solely fixes its magnitude. Now, imagine the ELBO as fixed and thus the magnitude of the combined tightness fixed, there is one degree of freedom left: the relative tightness of both of the variational bounds on mutual information. This is equivalent to saying that the mutual information may still vary in the range defined by the bounds. So, when optimising for the ELBO, we are not only not sensitive to rate-distortion trade-offs, but also not for the tightness ratio or the relative size of marginal KL and thus not for the mutual information in its range. Crucially, even if we were to fix the rate and distortion, this ratio may still vary. And, since, we argue this relative tightness is of importance for the goal of latent variable modelling, we think this is an important degree of freedom not accounted for in the rate-distortion plane from the work of Alemi et al. (2018a). We will expand on the consequences of this degree of freedom for optimisation in Section 4.2.2.

Connection to InfoVAE

When arriving at the conclusions in the previous Section 4.2, we have noticed this result is closely related to the motivation behind InfoVAE (S. Zhao et al., 2017) and its Lagrangian extension (S. Zhao et al., 2018). They motivate their objective with the argumentation that the VAE lacks a preference in mutual information between latent and observed variables under q_ϕ. They address this issue by adding it to the objective and, in the case of information maximisation mode, derive an objective that we wrote out in full in Equation 2.30. We can see that it contains the distortion, rate and a marginal KL term. So, by aiming at high mutual information we must, besides optimising the ELBO (increasing the *position of the range* of mutual information, the light grey area), also minimise the marginal KL (maximise the position of mutual information within that range, position of the black bar with circle on top within the light grey area). That is exactly what our analysis tells us. Recall that minimising marginal KL, as their information maximisation objective summons to do, is not trivial though because of the $q_\phi(\mathbf{z})$ term. This is the reason they turn to MMD as a proxy.

Although the derived result is similar, we note that there are a few important distinctions to be made between the work of S. Zhao et al. (2017) and S. Zhao et al. (2018) and ours, to start with the motivation. They arrive at their objective solely from an information maximisation perspective. We, however, reason from a generative modelling perspective and focus on analysis now where rate, distortion and thus the ELBO are already fixed. This makes that the range in which mutual information can exists is already defined by the bounds. We conclude that there are trade-offs to be made in such a situation between which bound on mutual information is loose. And we state that if we need to choose, a smaller marginal KL has our strict preference from an approximate inference point of view. This has the side effect of increased mutual information and the fact that it is a side-effect is crucial. If the bounds were not fixed, we would be able to increase MI perhaps at the *cost* of increasing marginal KL, a situation we would like to avoid at all times. This hints at autoencoder behaviour, or diminished capability to generalise outside of the data distribution and is thus undesirable.

4.2.1 Assessing the TransformerVAE along this axis for different optimisation techniques

In the previous section we came to the conclusion that marginal KL, or a preference on which mutual information bound gets loose, is not encoded in the ELBO. We also motivated that this quantity is important to us as it tells us how well our inference model q_ϕ is functioning with respect to our generative model p_θ. It indicates how efficient our code passed from encoder to the decoder is given a fixed decoder. If we achieve a model without a collapsed posterior, which means our decoder is exploiting the code to make predictions, we would like this to be as efficient as possible. In the previous section (Section 4.1), we have demonstrated the existence of the information preference problem in our model and that Free bits was the only method from the work by C. Li et al. (2020) that could counterbalance this preference and achieve substantial rate. We also noted that substantial rate is only an indication of information flowing from the encoder to the decoder, but does not give a complete image of the quality of the learned model. So, we will now assess different optimisation techniques along the newly proposed marginal KL axis in order to get more insights into the quality of models that converge at positive rates.

For our marginal KL analysis, we will focus on optimisation techniques that target a specific rate (for a review see the paragraph dedicated to methods that target a specific rate in Section 2.3.2). Concretely, we will compare 1) Free bits (FB), 2) Free bits with cyclical annealing (CYC-FB) and 3) Minimum Desired Rate (MDR) (Pelsmaeker & Aziz, 2019). We think those are good to compare as 1) appears in literature frequently and 2) is the method used by C. Li et al. (2020): analysing it might give us insight on the (added) effect of cyclical annealing. We added 3. since its strategy is rather different than that of Free bits. MDR is a method that seeks to maximise the ELBO while actively looking for solutions that satisfy the constraint of a minimum desired rate. Free bits, on the contrary, changes the objective to take away the incentive for the model to drive the rate down below a certain threshold as we discussed in Section 4.1.1. This entails that the criterion does not explicitly avoid finding solutions derived from invalid lower bounds. This happens for example when the rate stays below the target rate and the KL term is not subtracted from the distortion, leading to a potentially invalid lower bound. Additionally, since Free bits disregards the KL-divergence that is a below the target level, the risk exists for the KL-divergence appearing in the objective very little. This, in turn, causes no gradients to come from the KL-divergence term. For these reasons we think it is interesting to analyse these techniques side-by-side.

We have implemented MDR following the work of Pelsmaeker and Aziz (2019)[1]. Note that the target rate with this strategy is defined for all dimensions combined. To conveniently compare results to the per-dimension Free bits implementation, we report the MDR target rate divided by the number of dimensions.

We have highlighted before that marginal KL is a quantity that is intractable to compute and computationally expensive to estimate, due to the $\log q_\phi(\mathbf{z})$ term. A more pressing problem for analysis purposes, however, is is the empirical bound on the entropy:

$$0 \leq \log N - \log \sum_{j=1}^{N} q_\phi(\mathbf{z}_i|\mathbf{x}_j) \leq \log N \tag{4.12}$$

This causes the range of estimates that can be reliably made to be small and we thus seek for other approaches to 1) make a relative ordering between different optimisation techniques with respect to marginal KL and to 2) estimate the quantity for different optimisation techniques. But, before we move on to those approaches it a good idea to inspect the samples from models optimised with different optimisation strategies and target rates visually to get a sense of what they look like. In Figure 4.6 we plotted 1) a histogram of latent samples originating posteriors for randomly sampled data points 2) the probability density functions of a subset of those posteriors. To allow for visualisation, both are plotted marginally across a subset of 8 dimensions (light blue). As a reference, the probability density function of a standard Gaussian is added to the plots in orange.

From this figure, we can make a few observations. First, we can see that these plots confirm some basic intuition on how an autoencoder (AE) encodes information: as point estimates. The posteriors have vanished variance and they tend towards Dirac delta functions. Secondly, it confirms our view of a collapsed model (VAE) where all dimensions look exactly like the prior. Thirdly, we can notice some differences between different optimisation strategies (bottom three rows) and between a low target rate 0.25 (left column) and a higher target rate of 1.0 (right column). For all strategies we can see that the posterior variances are

[1] We made use of a Pytorch class implemented by Eelco van der Wel to perform constrained optimisation using Lagrange multipliers

decreased with a higher target rate. The probability density functions of the posteriors for all optimisation strategies look more like the prior's for a lower target rate. But, interestingly, MDR can be distinguished quite clearly from FB and CYC-FB for both target rates. Where models optimised with FB and CYC-FB seem to organise there dimensions rather similarly, MDR does not. Even with a low target rate of 0.25, MDR has two dimensions with posteriors that are visually quite different from the prior. Also for higher target rates the MDR optimised VAE models posteriors with different variance per dimension. For reference we also show a plot of a model optimised with only cyclical annealing and it confirms our view of its weak ability to achieve positive rate: the posteriors look still quite similar to the prior and it will thus be hard for the decoder to distinguish them.

One set of techniques to investigate the relative ordering in marginal KL are two sample test statistics. We can obtain a set of samples from $q_\phi(\mathbf{z})$ by first sampling from the dataset $\mathbf{x}_i \sim p_\mathcal{D}(\mathbf{x})$ and then sampling from the corresponding posterior $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}_i)$ (i.e. ancestral sampling). First, we perform a naive test on the two sets of samples marginally across dimensions, which reduces the two sample test from one in $\mathbb{R}^D$ to one in $\mathbb{R}^1$. For this we use the Kolmogorov-Smirnov (KS) test statistic (Smirnov, 1939; Kolmogorov, 1933). This test evaluates the distance between the two empirical cumulative distributions. On the left in Figure 4.7 the test statistic of this test is plotted for different optimisation techniques and target rates against the rate. As a second test we employ MMD (Gretton et al., 2012): a kernel based two sample test statistic. The MMD test statistic is plotted on the right side of Figure 4.7. For both tests holds, the larger the test statistic, the larger the deviation between the two sets of samples along the line of the test statistic.

From the plots we can observe a few things. First, both test statistics show similar trends and orderings between different optimisation techniques and target rates. Secondly, we can observe that Free bits shows higher test statistic values than the other two techniques and MDR is almost zero for both tests statistics. Recall that if the marginal KL is low, we have the desired effect of a tight upper bound on MI, which causes the rate to solely be determined by the mutual information. The codes are thus distinguishable for the decoder and while also respecting our generative modelling goals. We can also see that the VAE and the prior both have test statistics around zero. This is expected as the VAE is a collapsed model and its samples should look like those coming from the prior and those from the prior should naturally be very similar to those from the prior. A notable difference between the two plots is the relative positioning of the autoencoder (AE) objective. All in all, it is encouraging to see a similar pattern arise twice, but no definitive conclusions can be drawn from these tests. No definitive conclusions can be drawn because these statistical tests are designed to reject the null hypothesis of exact equivalence between two distributions $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$—which for finite sample sets is trivially false (Benavoli, Corani, Demšar, & Zaffalon, 2017)—and cannot conceive the possibility of approximate equivalence nor order alternatives by degree of approximate equivalence.

So, to further analyse the difference between samples from different models and the prior we turn to Bayesian analysis and train a Mixed Membership model (MM-model) (Airoldi, Blei, Erosheva, & Fienberg, 2014; Blei, 2014). In this model, groups of samples are modelled with learned components, which may be shared across groups (as opposed to regular mixture models where the components are group specific). The model is fitted with variational inference by specifying an approximate posterior family and optimising a lower bound on the model evidence (ELBO). The graphical model of the MM-models used for this analysis

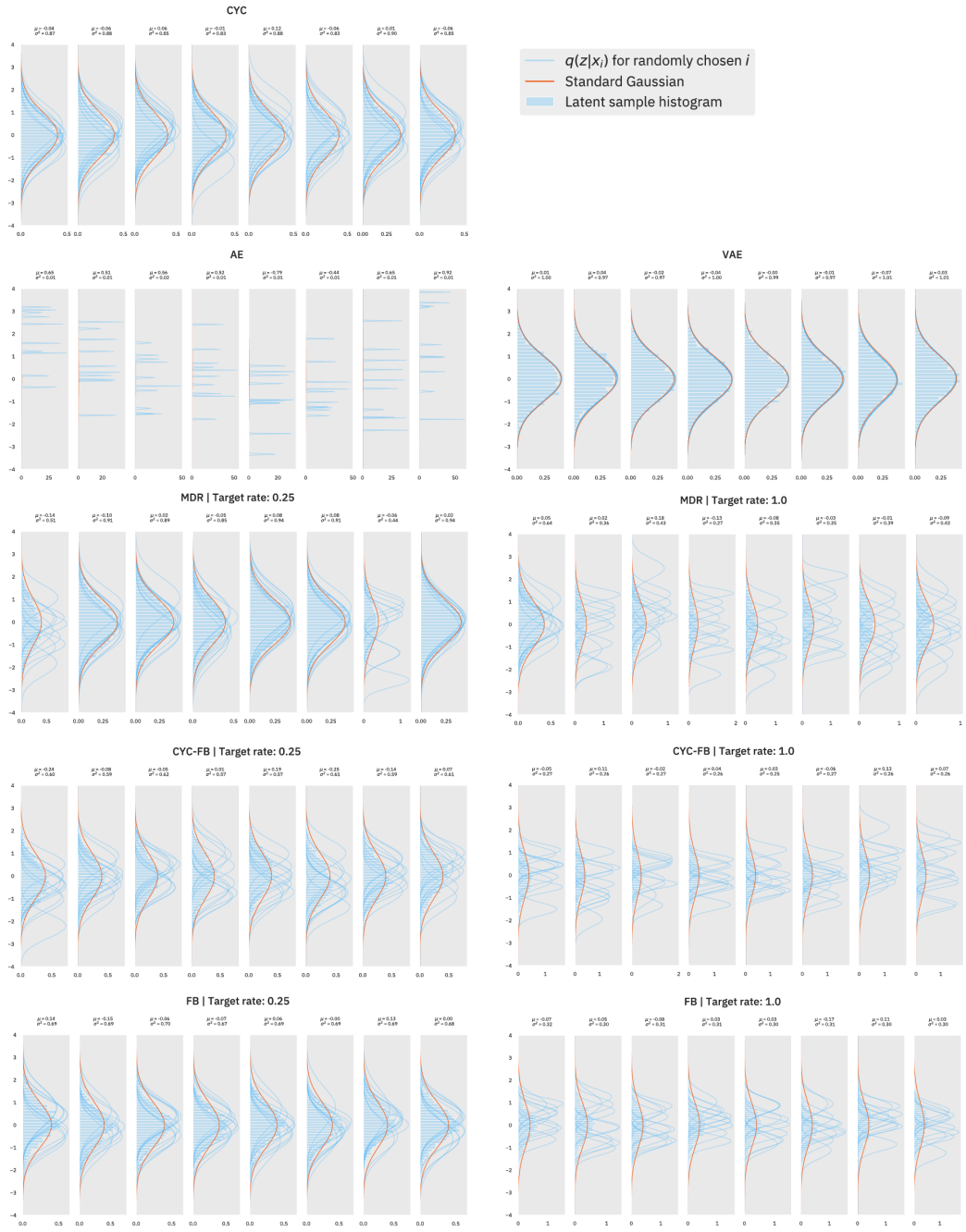

FIGURE 4.6: This plot visualises posteriors, samples from posteriors and the prior marginally across a subset of 8 dimensions for different optimisation techniques and target rates.

can be found in the Appendix B. Using this model for analysis is advantageous for the following reasons. First, it allows for univariate analysis as well as for the multivariate case. Secondly, it allows us to make a relative ordering with regards to the closeness of distributions $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$ that can serve as a proxy to marginal KL ordering. Thirdly, it relieves us from the limitations of null hypothesis testing. In the fourth place, it allows for analysing all groups simultaneously. That is, to evaluate samples from $q_\phi(\mathbf{z})$ from models that are optimised differently at once, not only pairwise against samples from the prior. And, finally, it is amenable to direct density estimation of $q_\phi(\mathbf{z})$.

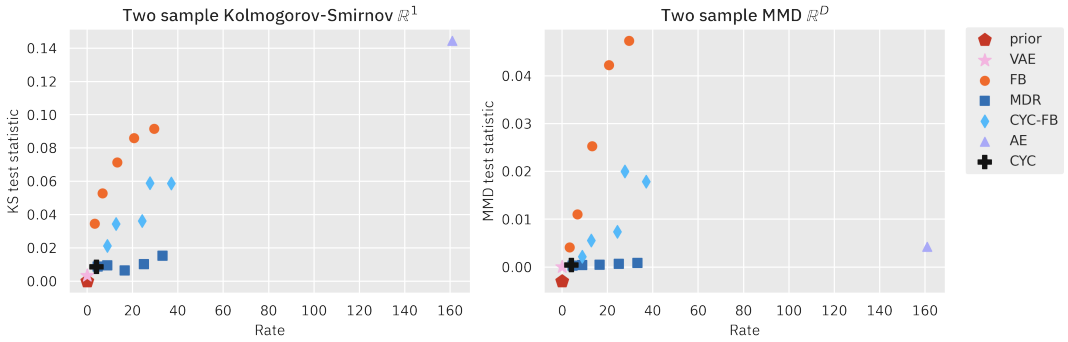

FIGURE 4.7: The Kolmogorov-Smirnov test statistic (left) and the MMD test statistic (right) plotted for samples from $q_\phi(\mathbf{z})$ from models optimised with different optimisation strategies and target rates versus samples from the prior $p(\mathbf{z})$

At the bottom of Figure 4.8 we can see the component distributions for different sample groups averaged over 1000 posterior samples for a MM-model fitted in $\mathbb{R}^D$ (left) and $\mathbb{R}^1$ (right). For both the $\mathbb{R}^1$ and $\mathbb{R}^D$ case, the prior can be modelled with one component only. This is expected as the prior is a one-component Gaussian with independent dimensions. The collapsed VAE model is modelled with the same, single component as the prior. Then, most interestingly, we can observe that all MDR models can be modelled with this same, single component. This agrees with the previous results of the KS and MMD test statistics that were low for MDR. The samples from $q_\phi(\mathbf{z})$ from a model optimised with MDR thus look a lot like those from the prior, hinting at low marginal KL. The other techniques, FB and CYC-FB are modelled with multiple different components. Only for low target rates we can observe the component of the prior appearing. This can be understood from the fact that these models are artificially kept from collapse with a small margin.

A natural use of the fitted MM-model is to employ it for density estimation and thus directly evaluate $\log q_\phi(\mathbf{z})$ to compute an estimate of $D_{KL}\left(q_\phi(\mathbf{z})||p(\mathbf{z})\right)$. In the top part of 4.8 we plotted the marginal KL estimates alongside the Index-code MI that is calculated by subtracting marginal KL from the rate as per the decomposition of Hoffman and Johnson (2016). In the marginal KL plot, we can see the same pattern appear as we observed in the KS and MMD plots with respect to all three optimisation techniques. We can see that for both Free bits and Free bits with cyclical annealing, marginal KL grows with increased rate, while for MDR it does not. Additionally, cyclical annealing seems to slow the growth in marginal KL with increased rate. In the mutual information plot, we can see that for all methods mutual information grows with the rate at a similar pace. This means that MDR is the only optimisation technique in our analysis that increases the rate without worsening our consistency between the two views of the joint or giving up on our goal of performing correct approximate inference.

Conclusion

In this section, we have analysed the existence and relative size of marginal KL for different optimisation strategies that target a specific rate. We have seen that all analyses techniques presented show similar relative orderings with respect to this quantity. This, on its own, is an encouraging result as it means that these are valuable tests to perform during training to keep track of marginal KL. Surprisingly, all analyses support evidence for the claim

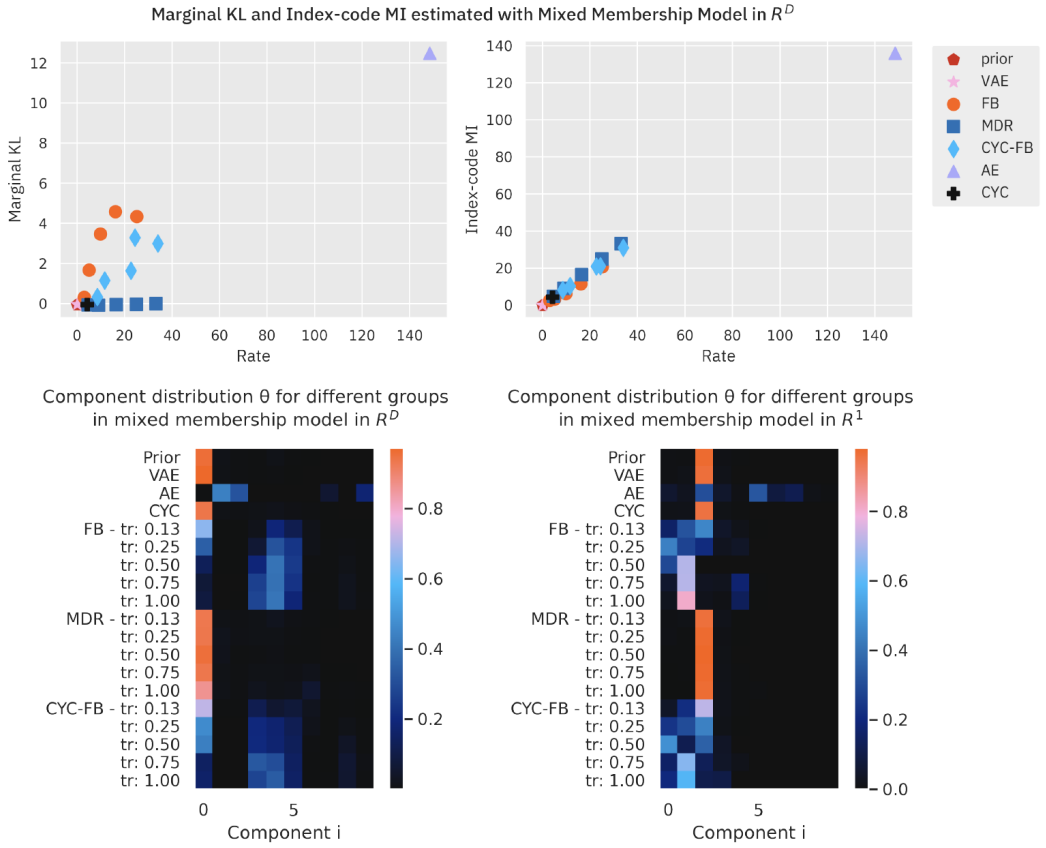

FIGURE 4.8: The bottom of this figure shows the component distribution for all sample groups averaged over 1000 posterior samples for a fitted Mixed Membership model in $\mathbb{R}^1$ (bottom-right) and in $\mathbb{R}^D$ (bottom-left). In the top the marginal KL (top-left) and Index-code MI $I_q(X;Z)$ are plotted against the rate, with $\log q_\phi(\mathbf{z})$ estimated under the same fitted $\mathbb{R}^D$ Mixed Membership model.

that MDR is the only optimisation strategy that can keep marginal KL low, while having increased rate and mutual information. Applying Free bits or Free bits with cyclical annealing, on the contrary, leads to models with increased rate and mutual information *at the cost of* increased marginal KL, which is an undesirable side effect when it comes to approximate posterior inference. So, all techniques that target a specific rate indeed achieve one of the intended goals of increasing mutual information, or the dependency between $\mathbf{z}$ and $\mathbf{x}$, but MDR is the only technique respecting the goals of variational inferenc (goal 3 and 4). This is interesting as there is a quite some research effort being invested increasing this quantity for generative latent variable models, with InfoVAE (S. Zhao et al., 2017, 2018) being the one we took a closer look at in this thesis. Our analysis shows that a shift in focus from increasing mutual information towards keeping marginal KL low while also achieving substantial rate is an alternative perspective that is more directly in line with the goal of performing good approximate inference.

4.2.2 Consequences for optimisation

In this section we will elaborate conceptually on what the consequences of our marginal KL analysis are for optimisation. In other words, we try to sketch a view of trade-offs made

regarding the ratio between the tightnesses of the bounds on mutual information while optimising the ELBO rather than for converged models (fixed combined tightness). This perspective in general, we argue, is rather underexposed also when considering the rate-distortion perspective by Alemi et al. (2018a) that reason on "tightest achievable bounds within a parametric family".

In section 4.2 we reasoned that for a fixed rate and distortion (and thus ELBO) the variational bounds on mutual information are fixed and thus the combined tightness of the lower and upper bound is fixed as well. We observed that the relative tightness and thus the relative size of the marginal KL may still vary in this situation. We used this observation for our analysis in the previous section: how do converged models with similar ELBO values differ in terms of marginal KL. Another interesting conceptual direction with a fixed ELBO, is optimisation. As we know that ELBO optimisation is only involved with finding better ELBO values, this marginal KL axis is another quantity ELBO optimisation is not sensitive to. Even when we are adopting techniques that push ELBO optimisation in a direction with higher rate solutions, we might still be tricked into areas that are pathological with regards to marginal KL.

Consider Figure 4.9 as a conceptual sketch of ELBO optimisation considering this additional axis of variation that we care about qualitatively. In the top right we draw a conceptual ELBO landscape as a function of our hypothetical parameters θ to explicitly appreciate the fact that during optimisation, we can only adapt our parameters in order to change the fitness of our model with respect to our objective (in this case the ELBO). This is important to realise when mapping other quantities of interest in a graph, such as the rate distortion plane (top left). Points that are adjacent in these visualisations, might actually lay far apart in parameter space and cannot be traded as easily as these visualisation might suggest.

In this figure we conceptualise three types of areas in the ELBO landscape, that differ *quantitatively* with respect to rate-distortion ratio and with respect to the relative tightness of the mutual information bounds and thus differ *qualitatively* with respect to our goals of representation learning and performing correct approximate posterior inference. Concretely, we distinguish the following three area types with the *same ELBO*, denoted with ■ , ✚ and ♦ in Figure 4.9:

1. ■ **(low rate, high distortion, low marginal KL)**: this an area where our posterior has collapsed. The rate has vanished as a consequence and our marginal KL is low as well because a small rate forces it to be by definition of the bounds and the fact that mutual information can not be negative. In this situation the distortion is high.

2. ✚ **(high rate, low distortion, low marginal KL)**: this is an area we ideally would like to end up in. We have reduced the distortion and make use of rate to do so, not at the cost of increasing marginal KL (this area has low marginal KL), but solely by increasing mutual information and using it efficiently to decrease distortion. We sketch this point as part of our parametric ELBO landscape, but we note is not known whether this point actually exists within it. The main problem of learning VAEs with strong decoders is that this point seems to be hard or impossible to find.

3. ♦ **(high rate, low distortion, high marginal KL)**: this an area where we have achieved low distortion with a high rate, but at the cost of increased marginal KL. This is the pathological area we wish to avoid as our approximate posterior inference is compromised. That is, our approximate posterior is not a good proxy to the true posterior. This

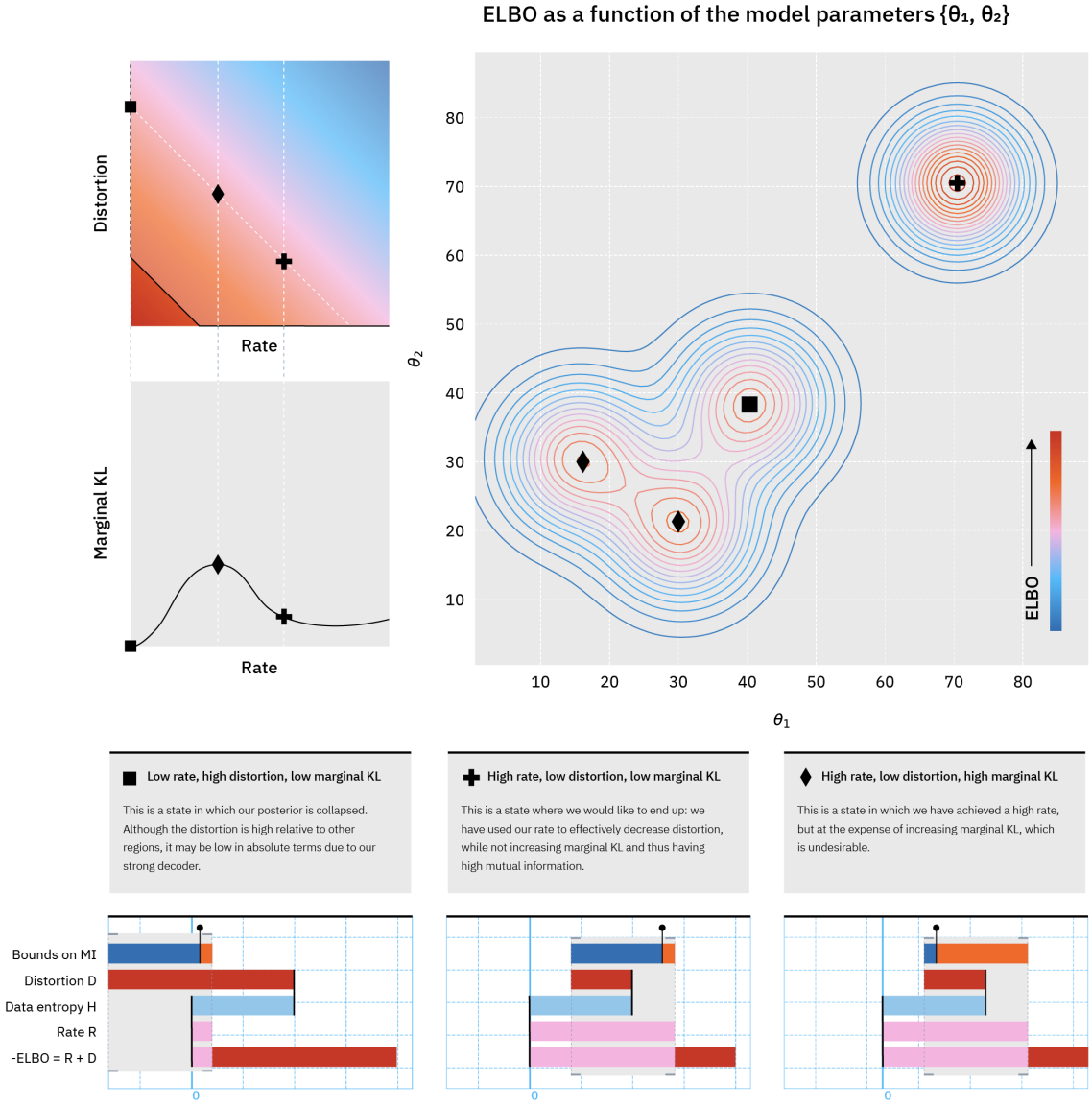

FIGURE 4.9: This figure shows the degree of freedom of marginal KL by sketching three area types in parameter space that are qualitatively different while having similar ELBO values

has consequences for the trustworthiness of the inference performed by the model and for the generalisability of our decoder. This area is drawn closer to the collapsed model (■) to highlight the fact that this are might be easier to reach from a collapsed point than the ideal point we would like to end up in (✚). Note that this area in fact has increased mutual information with respect to a collapsed state.

The left two plots highlight the fact that the rate-marginal KL should be seen as an addition to the rate-distortion perspective. The lower three diagrams connect the three area types visually to the diagram as presented in section 4.2.

Conclusion

In this section we sketched the possible consequences of a lack of preference with regards to marginal KL in ELBO optimisation. We hypothesise there might be areas in the ELBO

landscape where the rate and mutual information are increased at the cost of compromising our approximate inference. We also hypothesise that different areas in the ELBO landscape might be easier or harder to reach from a point of posterior collapse by explicitly drawing out parameter space. It also noteworthy that these pathological places potentially pose a specific hazard when actively seeking for higher rate solutions.

4.3 On architectural inductive biases

In the previous sections, we have looked into the optimisation of VAEs with strong decoders and techniques to circumvent a collapsed posterior. From the results in Section 4.1 we know we can learn models that make use of substantial rate with various techniques, such as Free bits or MDR. Even though some of these techniques may lead to undesirable side effects for variational inference, we know we can bring the VAE in a state where it starts to encode information and increase mutual information between the input and the latent representation. We also saw that increasing the rate, results in a drop in distortion (Figure 4.3, Section 4.1). Although this drop is not proportional to the increase in rate, indicating inefficient encoding, we have reason to believe the decoder utilises this information to decrease distortion, or the reconstruction loss. In this section, we investigate a natural follow-up investigation to explore *what kind* of information can be encoded by the TransformerVAE. Concretely, we will test what parts of the sequence are most affected by our encodings and link this to the *memory mechanism* (C. Li et al., 2020).

To assess the information encoded and the use of it by the decoder we will inspect differences between reconstructions given samples from the posterior distributions corresponding to these inputs and samples from prior. The former samples should carry information for those reconstructions as they are data point specific encodings $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}_n)$ and the latter are likely not to as they do not condition on $\mathbf{x}$: $\mathbf{z} \sim p(\mathbf{z})$. Note that even though the prior is not specific to one data point, it is still possible that by chance we sample a latent $\mathbf{z}$ that lays in the neighbourhood of $\mathbf{z}$ that originates from $q_\phi(\mathbf{z}|\mathbf{x}_n)$. We account for this that by sampling multiple latents per data point.

To measure the information difference between samples from the prior and from the posterior, we will use two metrics to quantify differences in output distributions: accuracy drop and Jenson-Shannon Divergence (JS-divergence) (Pelsmaeker & Aziz, 2019). Accuracy drop is a metric that compares the modes of the two distributions, while JS-divergence compares the distributions in full. Accuracy drop is a fit metric to get a quick indication of this behaviour because it is interpretable and easy to compute. This, unsurprisingly, is at the same time its shortcoming. Differences in distributions may be more subtle than mode shifts. This is where the JS-divergence can help out. JS-divergence is the symmetrised version of the KL-divergence and thus compares distributions in full in terms of expected information difference. We perform 5 experiments with samples from the prior and 5 samples from the posterior. We subtract the amount of internal variability within the 5 posterior experiments from the variability between the two sets.

In Figure 4.10 the mean of both metrics for different optimisation techniques and target rates are shown. From this figure we can make a few observations. First, the trend in both figures is very similar: higher target rates lead to an increase in both these metrics. Secondly, the autoencoder model shows the largest accuracy drop and JS-divergence between prior and posterior decodings. Both observations are not surprising as higher target rates cause

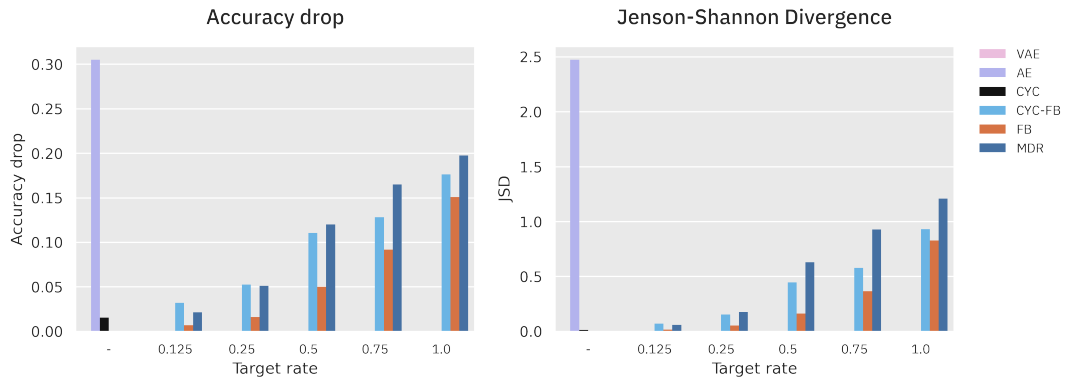

FIGURE 4.10: Average accuracy drop and Jenson-Shannon Divergence between output distributions of reconstructions using samples from the posterior and from the prior (averaged over 5 experiments, 3370 data points)

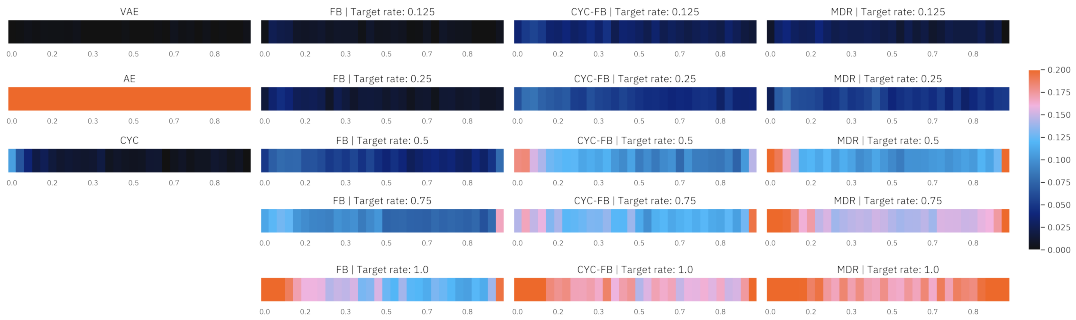

FIGURE 4.11: Accuracy drop between reconstruction predictions with prior and posterior samples (averaged over 5 experiments, 3370 data points)

the encoder to transmit more information. Additionally, between optimisation techniques, MDR consistently outperforms the other methods with respect to these metrics. Apparently, the feature encoding obtained by optimising with MDR is more efficient for the decoder to use and affects output distributions positively. Lastly, it seems that cyclical annealing in combination with Free bits also has a positive effect on these metrics compared to Free bits without cyclical annealing.

To get a more complete image of what kind of information is encoded, we additionally plot these metrics over relative sequence length. In Figure 4.11 we plot the accuracy drop

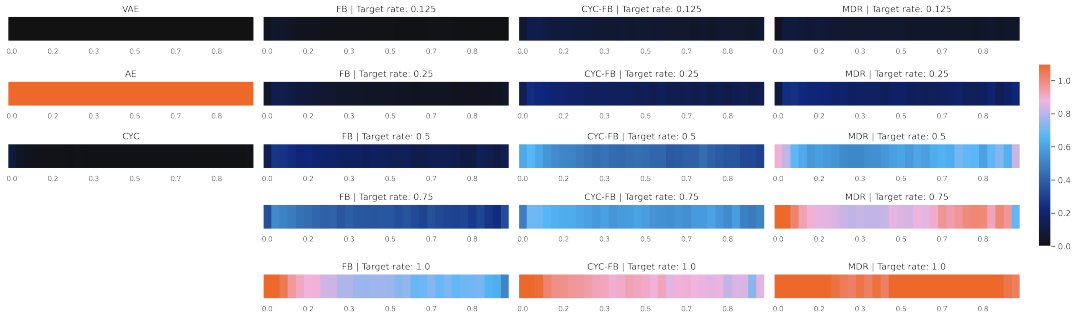

FIGURE 4.12: Jenson-Shannon Divergence between output distributions of text reconstructions with prior and posterior samples (averaged over 5 experiments, 3370 data points)

over relative sequence length and in Figure 4.12 we plot the JS-divergence over relative sequence length. Again, the metrics tell a similar story, although the JS-divergence plot is a bit more nuanced and smooth. In general we observe again, that for higher target rates the metrics are more affected. The figure also shows that for most techniques the beginning of the sequence is affected most. We can explain this by noticing that at the beginning of the sequence the information preference property is less strong because there is less context to affect output distributions locally with. Interestingly, the accuracy drop shows for a few settings (e.g. MDR with a target rate of 0.5, middle row, right-most column) an increase at the end of the sequence, suggesting the model encodes information regarding to the sentence length. This is a piece of information that can be spotted more easily by inspecting mode shift, rather than full distribution changes and thus shows up in the accuracy drop plot only.

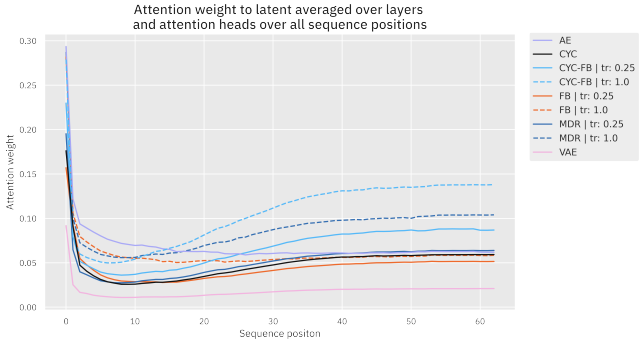

FIGURE 4.13: Attention weights assigned to the latent in the memory mechanism averaged over all heads and layers over the sequence (averaged over 3370 data points).

Lastly, to get a sense of how the decoder uses information stored in the latent representation, we inspect the attention weights assigned to the latent in the memory mechanism over the sequence, averaged over all heads and layers (Figure 4.13). We can see that, after a spike in the first token position, the attention drops to then stabilise again a bit higher later in the sequences. The pattern we observe here is somewhat similar to the accuracy drop plot (Figure 4.11) and JS-divergence plot (Figure 4.12). We hypothesise that if the encoded information is limited, and the features encoded by the model are shallow, the memory mechanism might bias towards encoding such a pattern, where mostly information on the beginning of the sequence is encoded. After all, attention will be stronger at the beginning of the sequence, when less context is present to attend to as well. In other words, the memory mechanism might emphasise the information property. From this perspective, it might be a worthwhile endeavour to design other mechanisms that induce different biases to encode features that can have a more holistic impact on the sequence and a substantial effect on the distortion that can justify the use of rate.

4.3.1 Conclusion

In this section we have inspected what kind of information can be encoded by the TransformerVAE by quantifying the effect of data point specific encodings on the decoder output distributions. We observed that the metrics used, agree with the mutual information trend plotted in Section 4.2.1. Thus, we have reason to believe that information is, besides being transmitted to, also in fact utilised by the decoder. Plotting the effect of the encodings on the

predictions over the sequence agrees with our view that the features encoded are rather shallow. We observe a tendency to store information on the beginning of the sequence, where the information preference property is the least strong. Additionally, accuracy drop reveals that some models seem to encode information on the length of the sequence. Lastly, when inspecting the attention weight pattern associated with the latent representation, we notice that this mechanism might actually induce biases that refrain the model from encoding more holistic and efficient features. Further research might design architectural modifications concerning the mechanism to add the latent to the decoder that cater these characteristics we would like our encodings to possess more directly, to help overcome the strong decoder problem from an architectural point of view.

Chapter 5

Conclusion & future research

In this concluding chapter, we will give a summary of the most important findings of this thesis and hint at future research. We will follow the structure as outlined in the Introduction and as given substance to in Chapter 4. But, first, we will repeat the qualitative goals we argue should be in balance when optimising a VAE in the context of powerful density estimators, especially when posterior collapse remains a challenge. We specify these goals to be: 1) generative modelling capabilities, 2) dependency between observed and unobserved variables, 3) consistency between the two views of the joint and 4) the quality of our approximate inference model. In this thesis research we have aimed to provide an analysis that is complementary to common existing modes of analysis, with an emphasise on the fourth goal: the quality of approximate inference.

Demonstrating the information preference problem: a TransformerVAE is a *very* strong decoder. In Section 4.1, we have argued and demonstrated a large, pre-trained Transformer model to be a very strong decoder and how this leads to the information preference property. This helps to align on what we think should exactly be the goal of this line of research: understand how to use strong decoders while also satisfying the goals of representation learning. Additionally, it helps understand the relative effectiveness of various techniques to combat posterior collapse, such as Free bits that directly addresses the information preference property versus cyclical annealing that (in our setting) biases towards higher rate solutions only lightly. This is practical as the literature on posterior collapse is diverse in modelling settings (e.g. weak versus strong decoders), data domain (e.g. text versus images) and architectural set-ups (e.g. recurrent architectures versus Transformers). Moreover, we inspected the rate-distortion plane to observe typical patterns that clearly show the effect of the information preference problem: the strong decoder does not leave for enough high impact features for the encoder to be found and transmitted to the decoder to lower distortion proportionally.

A missing axis in the information theoretic view of VAEs: the quality of approximate posterior inference. Subsequently, in Section 4.2, we explore an information theoretic view to come to the conclusion there is a quantity (i.e. marginal KL) that is not explicitly accounted for in the rate-distortion view by Alemi et al. (2018a) while being directly relevant to the quality of approximate inference. Concretely, it is a dimension that might still vary even when fixing both the ELBO *and* the rate-distortion ratio. We argue that the goal of performing accurate approximate posterior inference is important because it is both underexposed in related work and likely to be compromised in the context of strong decoders

that tend to overcompensate in the VAE. When analysing existing methods to target a specific rate with the intention of de-collapsing the *true* posterior we find notable differences between the prevalent method of Free bits (Kingma et al., 2016) and the more recently proposed method of MDR (Pelsmaeker & Aziz, 2019). Where both methods achieve higher mutual information, higher rate solutions found by Free bits are compromised with increased marginal KL, while MDR manages to achieve high rate purely consisting of mutual information while also matching the marginals $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$. This stresses the importance in a nuanced difference between the findings of J. Zhao et al. (2018) and S. Zhao et al. (2017) and ours: rather than focusing on maximising mutual information we might shift our focus to minimising marginal KL. Our analysis leads to two practical recommendations. First, we recommend to avoid using objectives that violate the ELBO, such as Free bits but also e.g. β-VAE with $\beta < 1$. And, secondly, we recommend to conduct (simple) statistical or visual checks during training to see if there is an indication of hazardous marginal KL increase. We have demonstrated the use of multiple techniques that can help diagnose this and found the Mixed Membership model a valuable addition to our analysis toolbox, with the added advantage of being able to directly perform density estimation of $q_\phi(\mathbf{z})$. We closed of this section by drawing out conceptual areas in the ELBO landscape along parametric axes, to appreciate potential pitfalls regarding this degree of freedom during optimisation. This thesis research leaves investigations of (practical) implications of compromised approximate inference for future research. We think it would be a worthwhile undertaking to, for example, explore useful descriptive statistics to systematise implications increased marginal KL may have on the (generated) sample space, for which Bayesian analysis models such as the Mixed Membership model are a great tool.

On architectural inductive biases. In the last section of our main chapter (Section 4.3), we have investigated what kind of information can be encoded in the latent representation by the TransformerVAE. The rate-distortion plane provided evidence for the claim that our encodings are not optimally efficient, because rate could not be used to lower distortion with equal proportions. This section illustrates this inefficiency by hinting at shallow feature encoding. The accuracy drop and JS-divergence plots reveal patterns that the encodings mostly affect predictions in the beginning of the sequence and perhaps the mode of the distribution at very end of the sequence (suggesting sequence length encoding). If we were to encode efficient, high-level and global features (such as topic or sentiment), we expect these patterns to be more distributed over the entire sequence as the features would have a more holistic impact. We further speculated on the role of the memory mechanism in the observed pattern and note this could actually further enforce the information preference property, while we would like it to be counteracted. Thus, we think that architectural inductive biases should be considered carefully when designing mechanisms to add the latent code to the decoder and are worth devoting future research to, especially now this new architectural model class of the Transformer has arrived.

Appendix A

Performance statistics all runs

A.1 Penn Treebank

	Optimisation	Memory	Embedding	Drop-out	ELBO	Rate	Distortion	I.W. Perplexity	Accuracy
0	AE	False	True	0.00	-144.28	43.14	101.14	228.87	0.33
1	AE	True	False	0.00	-106.75	5.03	101.72	46.89	0.33
2	AE	True	True	0.00	-234.10	163.55	70.54	20120.17	0.51
3	CYC-FB-0.5	True	False	0.00	-99.47	4.00	95.48	36.71	0.35
4	CYC-FB-0.5	True	False	0.40	-104.44	3.50	100.94	44.56	0.32
5	CYC-FB-0.5	True	True	0.00	-103.91	14.66	89.26	39.53	0.38
6	CYC-FB-0.5	True	True	0.40	-108.57	20.91	87.66	46.62	0.39
7	DEC-ONLY	False	False	0.00	-	-	96.36	-	-
8	MDR-0.5	True	False	0.00	-105.25	16.26	88.99	41.79	0.42
9	MDR-0.5	True	False	0.40	-99.70	17.17	82.53	34.71	0.42
10	MDR-0.5	True	True	0.00	-102.65	16.66	85.99	37.53	0.42
11	MDR-0.5	True	True	0.40	-99.47	16.30	83.17	34.20	0.42
12	VAE	True	False	0.00	-94.39	0.00	94.39	33.57	0.36
13	VAE	True	False	0.40	-94.63	0.00	94.62	33.91	0.35
14	VAE	True	True	0.00	-94.43	0.01	94.42	33.24	0.36
15	VAE	True	True	0.40	-94.32	0.00	94.32	33.32	0.36
16	VAE	True	True	0.60	-95.78	0.21	95.57	34.92	0.35
17	VAE	True	True	0.80	-101.74	5.50	96.24	40.34	0.34

TABLE A.1: Global performance statistics on the runs we performed on Penn Treebank dataset

A.2 Yelp Review dataset

	Optimisation	Memory	Embedding	Drop-out	ELBO	Rate	Distortion	I.W. Perplexity	Accuracy
0	AE	False	True	0.00	-422.04	300.38	121.66	2746.90	0.56
1	AE	True	False	0.00	-453.88	337.99	115.89	4819.74	0.57
2	AE	True	True	0.00	-402.97	285.97	117.01	1726.43	0.57
3	CYC-FB-0.5	True	False	0.00	-191.96	29.55	162.41	24.66	0.43
4	CYC-FB-0.5	True	False	0.40	-201.70	41.41	160.29	28.64	0.44
5	CYC-FB-0.5	True	True	0.00	-192.38	24.38	168.00	25.03	0.42
6	CYC-FB-0.5	True	True	0.40	-200.75	40.05	160.71	28.30	0.44
7	DEC-ONLY	False	False	0.00	-	-	183.43	-	-
8	MDR-0.5	True	False	0.00	-182.89	16.52	166.38	21.73	0.42
9	MDR-0.5	True	False	0.40	-188.70	16.69	172.01	23.62	0.40
10	MDR-0.5	True	True	0.00	-183.08	16.51	166.58	21.68	0.42
11	MDR-0.5	True	True	0.40	-187.32	16.81	170.51	23.09	0.41
12	VAE	True	False	0.00	-179.86	0.00	179.86	21.67	0.38
13	VAE	True	False	0.40	-184.12	0.00	184.12	23.29	0.37
14	VAE	True	True	0.00	-179.92	0.00	179.92	21.67	0.38
15	VAE	True	True	0.40	-185.08	0.34	184.74	23.52	0.37

TABLE A.2: Global performance statistics on the runs we performed on Yelp Review dataset

Appendix B

Mixed Membership Model graphical model

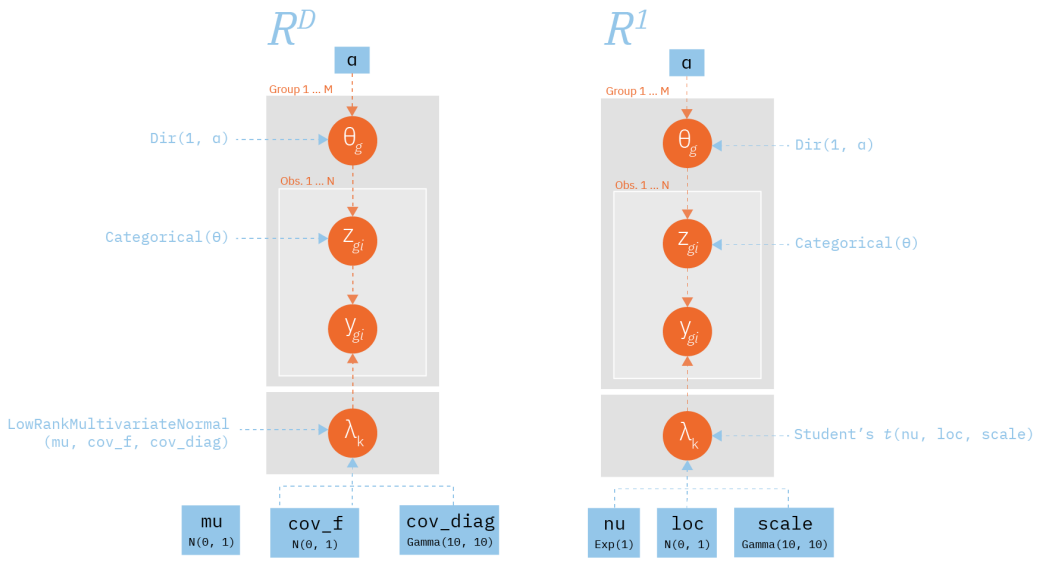

FIGURE B.1: Graphical models of the Mixed Membership Models used for analysis in Section 4.2.1

In Figure B.1 we draw out the graphical models with corresponding families of distributions to model each random variable in the Mixed-Membership models used for the analysis in Section 4.2.1. The Mixed-Membership model induces a number of distributions (a.k.a. components) of a given family (e.g. Student's t) to explain all observed data and these components may be shared between groups within the data (e.g. latents originating from models optimised with different optimisation techniques). The data generative process is assumed to be the following. For each group, a component distribution is sampled from a shared Dirichlet prior distribution. This distribution parameterises a draw from a categorical for each observation in the group to determine which component is used draw the observation from. Observations from different groups may thus be drawn from a shared component. Posterior inference is approximated via stochastic variational inference (Hoffman, Blei, Wang, & Paisley, 2013) using Pyro (Bingham et al., 2019). The posterior distribution over the latent

parameters of the analysis model (i.e. components and mixing coefficients) can be used to compare the different groups and to estimate the density of samples from each group.

References

Airoldi, E. M., Blei, D. M., Erosheva, E. A., & Fienberg, S. E. (2014). *Handbook of mixed membership models and their applications*. doi: 10.1201/b17520

Alemi, A. A., Poole, B., Fische, I., Dillon, J. V., Saurous, R. A., & Murphy, K. (2018a). Fixing a broken elbo. *35th International Conference on Machine Learning, ICML 2018*, *1*(1), 245–265.

Alemi, A. A., Poole, B., Fische, I., Dillon, J. V., Saurous, R. A., & Murphy, K. (2018b). Fixing a broken elbo. *35th International Conference on Machine Learning, ICML 2018*, *1*, 245–265.

Benavoli, A., Corani, G., Demšar, J., & Zaffalon, M. (2017). Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis. *Journal of Machine Learning Research*, *18*.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. doi: 10.1109/TPAMI.2013.50

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., … Goodman, N. D. (2019). Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, *20*.

Blei, D. M. (2014). *Build, compute, critique, repeat: Data analysis with latent variable models* (Vol. 1). doi: 10.1146/annurev-statistics-022513-115657

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*(4-5). doi: 10.1016/b978-0-12-411519-4.00006-9

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2016). Generating sentences from a continuous space. In *Conll 2016 - 20th signll conference on computational natural language learning, proceedings* (pp. 10–21). doi: 10.18653/v1/k16-1002

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). Language models are few-shot learners. *arXiv*.

Burda, Y., Grosse, R., & Salakhutdinov, R. (2016). Importance weighted autoencoders. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 1–14.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). Understanding disentangling in β-VAE. *arXiv*(Nips).

Chen, T. Q., Li, X., Grosse, R., & Duvenaud, D. (2018). Isolating sources of disentanglement in variational autoencoders. *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*(NeurIPS).

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., … Abbeel, P. (2017). Variational lossy autoencoder. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 1–17.

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv*.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. V., & Salakhutdinov, R. (2020). Transformer-XL: Attentive language models beyond a fixed-length context. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2978–2988. doi: 10.18653/v1/p19-1285

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete

Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1). doi: 10.1111/j.2517-6161.1977.tb01600.x

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*(Mlm), 4171–4186.

Dieng, A. B., Kim, Y., Rush, A. M., & Blei, D. M. (2019). Avoiding latent variable collapse with generative skip models. In *Proceedings of machine learning research* (Vol. 89, pp. 2397–2405). PMLR.

Fang, L., Li, C., Gao, J., Dong, W., & Chen, C. (2019). Implicit deep latent variable models for text generation. *arXiv*, 3946–3956.

Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., & Carin, L. (2019). Cyclical annealing schedule: A simple approach to mitigating KL vanishing. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 240–250. doi: 10.18653/v1/N19-1021

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, *13*, 723–773.

Hartley, H. O. (1958). Maximum Likelihood Estimation from Incomplete Data. *Biometrics*, *14*(2). doi: 10.2307/2527783

Higgins, I., Matthey, L., Arka, P., Burgess, C., Glorot, X., Botvinick, M., … Lerchner, A. (2016). β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *Urban Affairs Review*, *44*(6), 807–831. Retrieved from https://openreview.net/forum?id=Sy2fzU9gl doi: 10.1177/1078087408328050

Hochreiter, S., & Urgen Schmidhuber, J. (1997). Long Shortterm Memory. *Neural Computation*, *9*(8), 17351780. Retrieved from http://www7.informatik.tu-muenchen.de/~hochreit%0Ahttp://www.idsia.ch/~juergen

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, *14*, 1303–1347.

Hoffman, M. D., & Johnson, M. J. (2016). Elbo surgery: yet another way to carve up the variational evidence lower bound. *Workshop in Advances in Approximate Bayesian Inference, NIPS*, *1*, 2.

Hooker, S. (2020). The Hardware Lottery. Retrieved from http://arxiv.org/abs/2009.06489

Jordan, M. I., & Saul, L. K. (1999). *An Introduction to Variational Methods for Graphical Models* (Vol. 37; Tech. Rep.).

Kim, Y., Wiseman, S., & Rush, A. M. (2018). A Tutorial on Deep Latent Variable Models of Natural Language. , 1–48. Retrieved from http://arxiv.org/abs/1812.06834

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*(Nips), 4743–4751.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*(Ml), 1–14.

Kingma, D. P., & Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, *12*(4), 307–392. doi: 10.1561/2200000056

Kolmogorov, A. (1933). Sulla determinazione empirica di una lgge di distribuzione. *Inst. Ital. Attuari, Giorn.*, *4*, 83–91.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. , 1–17. Retrieved from http://arxiv.org/abs/1909.11942

Li, B., He, J., Neubig, G., Berg-Kirkpatrick, T., & Yang, Y. (2020). A surprisingly effective fix for deep latent variable modeling of text. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 3603–3614.

Li, C., Gao, X., Li, Y., Li, X., Peng, B., Zhang, Y., & Gao, J. (2020). Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. Retrieved from http://arxiv.org/abs/2004.04092

Liu, D., & Liu, G. (2019). A Transformer-Based Variational Autoencoder for Sentence Generation. In *2019 international joint conference on neural networks (ijcnn)* (Vol. 2019-July, pp. 1–7). IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/8852155?casa_token=_0OF7tROKnAAAAAA:2otr1aU5lYQHRlKMOGtSKBP1w7rtZn647yUA9pOOYruwivvY-rSLi30foT8VaWIeDKgfCqR30Q doi: 10.1109/IJCNN.2019.8852155

Liu, Q., Chen, Y., Chen, B., Lou, J. G., Chen, Z., Zhou, B., & Zhang, D. (2020). You impress me: Dialogue generation via mutual persona perception. *arXiv*. doi: 10.18653/v1/2020.acl-main.131

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. (1). Retrieved from http://arxiv.org/abs/1907.11692

Logan IV, R. L., Gardner, M., & Singh, S. (2020). On Importance Sampling-Based Evaluation of Latent Language Models. , 2171–2176. doi: 10.18653/v1/2020.acl-main.196

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial Autoencoders. Retrieved from http://arxiv.org/abs/1511.05644

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational linguistics - Association for Computational Linguistics (Print)*, *19*(2).

Mathieu, E., Rainforth, T., Siddharth, N., & Teh, Y. W. (2019). Disentangling disentanglement in variational autoencoders. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 7744–7754.

Mihaylova, T., Niculae, V., & Martins, A. F. T. (2020). Understanding the Mechanics of SPIGOT: Surrogate Gradients for Latent Structure Learning.. doi: 10.18653/v1/2020.emnlp-main.171

Pelsmaeker, T., & Aziz, W. (2019). Effective Estimation of Deep Generative Language Models. *arXiv*(1). doi: 10.18653/v1/2020.acl-main.646

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Improving Language Understanding by Generative Pre-Training. *OpenAI*, 1–10. Retrieved from https://gluebenchmark.com/leaderboard%0Ahttps://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. ,

21, 1–67. Retrieved from http://arxiv.org/abs/1910.10683

Razavi, A., Vinyals, O., Van Den Oord, A., & Poole, B. (2019). Preventing posterior collapse with δ-VAES. *7th International Conference on Learning Representations, ICLR 2019*.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *31st International Conference on Machine Learning, ICML 2014*, *4*, 3057–3070.

Roberts, A., Engel, J., Raffel, C., Hawthorne, C., & Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music. *35th International Conference on Machine Learning, ICML 2018*, *10*, 6939–6954.

Rothe, S., Narayan, S., & Severyn, A. (2020). Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. *Transactions of the Association for Computational Linguistics*, *8*, 264–280. doi: 10.1162/tacl{_}a{_}00313

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In *54th annual meeting of the association for computational linguistics, acl 2016 - long papers* (Vol. 3). doi: 10.18653/v1/p16-1162

Smirnov, N. V. (1939). Estimate of deviation between empirical distribution functions in two independent samples. *Bulletin Moscow University*, *2*(2), 3–16.

Titsias, M. K., & Lázaro-Gredilla, M. (2014). Doubly stochastic variational bayes for non-conjugate inference. In *31st international conference on machine learning, icml 2014* (Vol. 5).

Tomczak, J. M., & Welling, M. (2017). VAE with a vampprior. *arXiv*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), 5999–6009.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., & Titov, I. (2019). Analyzing multi-head self-attention:Specialized heads do the heavy lifting, the rest can be pruned. *arXiv*, 5797–5808.

Wu, X., & Lode, M. (2020). Language Models are Unsupervised Multitask Learners (Summarization). *OpenAI Blog*, *1*(May), 1–7. Retrieved from https://github.com/codelucas/newspaper

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, *32*, 1–18.

Yang, Z., Hu, Z., Salakhutdinov, R., & Berg-Kirkpatrick, T. (2017). Improved variational autoencoders for text modeling using dilated convolutions. *34th International Conference on Machine Learning, ICML 2017*, *8*, 5917–5928.

Zhang, X., Zhao, J., & Lecun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* (Vol. 2015-Janua).

Zhao, J., Kim, Y., Zhang, K., Rush, A., & LeCun, Y. (2018). Adversarially regularized autoencoders. In *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 5902–5911). PMLR. Retrieved from http://proceedings.mlr.press/v80/zhao18b.html

Zhao, S., Song, J., & Ermon, S. (2017). InfoVAE: Information Maximizing Variational Autoencoders. Retrieved from http://arxiv.org/abs/1706.02262

Zhao, S., Song, J., & Ermon, S. (2018, 6). The Information Autoencoding Family: A Lagrangian Perspective on Latent Variable Generative Models. Retrieved from http://

arxiv.org/abs/1806.06514