EPFL

Thesis n° 11133

Nonparametric Variational Information Bottleneck: Attention-based Architectures as Latent Variable Models

Presented on 5th September 2025

School of Engineering
L'IDIAP Laboratory
Doctoral program in Electrical Engineering

for the award of the degree of Docteur ès Sciences (PhD)

by

Fabio James FEHR

Accepted on the jury's recommendation

Prof. V. Cevher, jury president
Prof. D. Gatica-Perez, Dr J. Henderson, thesis directors
Dr J. Tomczak, examiner
Dr J. Hensman, examiner
Prof. N. Flammarion, examiner

École
polytechnique
fédérale
de Lausanne

2025

I can't promise it will be easy, but it will be so cool!
— Dr. J. Henderson, during my hiring interview

A PhD, like happiness, cannot be pursued; it must ensue – a byproduct of
curiosity, passion, and the courage to pursue what is cool:
that type of cool which excites and inspires you.
— V. Frankl-inspired

I dedicate my pursuit of cool research
to all the lovely people who supported me . . .

Acknowledgements

If you're reading this and knew me between 2021-2025, you undoubtedly played an important role in my journey. Know I'm grateful for you.

First, I want to thank my supervisor, **Jamie**, for believing in me and taking a chance on a long-haired nobody from South Africa. I am grateful to have been part of the NVIB adventure, and I learned a great deal about meaningful research along the way. Your passion for research, commitment to your students, and your pursuit of ambitious ideas have left a lasting impression on me. These values will shape how I approach what comes next. I am proud to be one of your alumni. Thank you for inspiring me. Long live NVIB!

I would like to thank my thesis committee for their time, support, and thoughtful engagement throughout this journey: Prof Volkan Cevher, Prof Nicolas Flammarion, Prof Daniel Gatica-Perez, Dr Jakub M. Tomczak, and Dr James Hensman. Daniel, thank you for your steady presence and big-picture perspective. Volkan and Nicolas, I appreciated your continued support from the candidacy exam through to the final defence. James, thank you for travelling to Switzerland and for your generous reflections and advice. Jakub, I am grateful for your thoughtful and encouraging questions during the defence. I am thankful to each of you.

Family A big part of this PhD journey was connecting with my family in Switzerland on my father's side, who passed away when I was young. I am lucky to have such warm Swiss family (my brother, aunts, uncles, and the sea of cousins) who welcomed me with open arms, even when I learned French instead of German (je suis désolé, aber ich werde es versuchen! oder?). When the journey was dark and cold (most winters), I was rejuvenated by all the sunshiney family who remained in South Africa. A special mention to my brothers, Cei and Anson, and especially to my mother, Mary. Mum, your wisdom and love kept me grounded, and your words of pride meant more than I can say. I hope Dad would be proud too. A heartfelt appreciation to Nacho, my beloved doggo, for his unconditional love and understanding no matter how long I was gone. May you rest in peace, my boy. Lastly, thank you to Bianca and the Rooseboom family for the loving support and acceptance of me and my crazy academic escapades.

Friends This one was tough to write. There are so many people, memories, and moments I could mention, but I'll start with the crazy multicultural band of misfit researchers who crawled out of Visp: **Idiapers**. I'm not going to do the typical thing of listing names, because that always leaves out people who mattered. The ones who smiled and bonjoured

in the corridor. The ones who remembered birthdays. The ones who made weekend deadline grinds feel a little brighter. The ones who organised events even when no one arrived. Beyond my close friends, who should be mentioned by name (you know who you are..), these are the people who make Idiap special. This community never misses a chance to celebrate. We collaborated on wild ideas, skied, hiked, played squash, swam at the pool with the grannies, went to concerts in Montreux, ran marathons, gardened in the office, played music, travelled for conferences and had holidays, learned each others languages (mostly swearing), cooked experimental meals, and partied late on week nights. I will remember fondly the late-night train ramblings and coffee-table chats about how we could change the world with AI. I was lucky to have been a small part of it this silly community. Thank you. **Andrei**, my brother in PhD, housemate, and dear friend, you deserve a special mention. The problem solving, cooking, synth music, silly marathons, and above all, I respect your values of kindness and inclusivity. To my long-time friends in South Africa and now scattered far and wide, thank you for always loving, soeking and accepting me, making it feel like no time has passed.

Figure 1: A gratitude video I created for all the friends!

The NVIBible Before thee lies the *NVIBible*, sacred manuscript of the Nonparametric Variational Information Bottleneck. In thy first year, the *Not Very Informed Boy* wandered the wilderness of Bayesian theory. The second was full of rigour, with *Nervously Verifying Initial Beliefs*, casting hope upon the maths and the code. In the third came *New Visions, Innovative Breakthroughs*, yet the conference scrolls returned inscribed: *Novel, Very Interesting, But...* The fruit of results was meagre, and the elders judged: *Not Very Impressive Boasts*. And cometh the fourth year, lo, it became *Now Various Iterations. Building!*. Jamie, our prophet, beheld the light. Together with I, the scholar of the gospel, preacher, and sometime heretic scribed the *new* new testament. Go forth disciples and study the divine NVIB prophecy, for it may bring thee insight, or even better, joy and mild amusement.

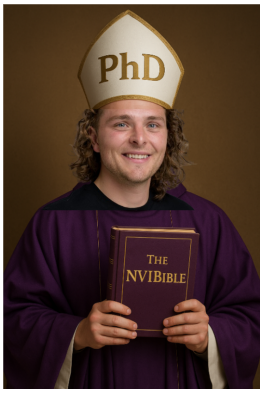


Figure 2: Scholar Fabio: **P**reacher of the **h**idden **D**istributions.

Abstract

Transformers have achieved remarkable success across modalities including text, graphs, speech, and vision, enabled by the attention mechanism. Yet the inductive biases that shape how attention encodes information and supports generalisation are still not well understood. Latent variable models offer a principled framework for explaining the encoded information, improving generalisation through regularisation, and enabling generative modelling. However, applying latent variable models to attention-based architectures is challenging, as attention functions over sets that are both variable in size and permutation-invariant. This thesis introduces the Nonparametric Variational Information Bottleneck (NVIB), a deep latent variable framework that models attention as posterior inference over a Dirichlet process mixture, aligning naturally with these set-based properties. We show that NVIB enables training a novel Transformer-based variational autoencoder from scratch, sparsifying the number of embeddings while regularising their content. As a generative model, it supports smooth interpolation and sampling within variable-sized latent spaces. When applied across stacked self-attention layers, NVIB induces hierarchical abstraction, improving interpretability, robustness, and linguistic alignment. This framework allows for pretrained Transformers to be reinterpreted as nonparametric variational models. NVIB reveals how they encode and separate reliable from unreliable information, enabling a novel and controllable post-training regularisation that improves out-of-distribution generalisation. Finally, NVIB boosts out-of-distribution performance during fine-tuning on speech, text, graph, and vision benchmarks, confirming its effectiveness in inducing generalisable representations across diverse models and tasks. Overall, the thesis offers a variational Bayesian perspective on attention, unifying regularisation, explanation, and generation, and opening new paths for advancing representation learning.

Key words: Transformers, Attention, Latent Variable Models, Variational Inference, Nonparametric Bayesian Methods, Dirichlet Processes, Information Bottleneck, Representation Learning, Generalisation, Regularisation, Interpretability, Abstraction.

Résumé

Les Transformers ont connu un succès remarquable dans plusieurs modalités, notamment le texte, les graphes, la parole et la vision, grâce au mécanisme d'attention. Pourtant, les biais inductifs qui déterminent la manière dont l'attention encode l'information et favorise la généralisation restent mal compris. Les modèles à variables latentes offrent un cadre rigoureux pour expliquer l'information encodée, améliorer la généralisation via la régularisation, et permettre la modélisation générative. Cependant, l'application de modèles à variables latentes aux architectures basées sur l'attention est difficile, car l'attention opère sur des ensembles à la fois de taille variable et invariants par permutation. Cette thèse introduit le Nonparametric Variational Information Bottleneck (NVIB), un cadre profond à variables latentes qui modélise l'attention comme une inférence a posteriori sur un mélange de processus de Dirichlet, s'alignant naturellement avec ces propriétés d'ensemble. Nous montrons que NVIB permet d'entraîner un autoencodeur variationnel inédit basé sur un Transformer, en réduisant le nombre d'embrayeurs tout en régularisant leur contenu. En tant que modèle génératif, il permet une interpolation fluide et un échantillonnage dans des espaces latents de taille variable. Lorsqu'il est appliqué à plusieurs couches d'auto-attention empilées, NVIB induit une abstraction hiérarchique, améliorant l'interprétabilité, la robustesse et l'alignement linguistique. Ce cadre permet également de réinterpréter les Transformers préentraînés comme des modèles variationnels non paramétriques. NVIB révèle comment ils encodent et séparent l'information fiable de l'information non fiable, permettant une régularisation post-entraînement nouvelle et contrôlable, qui améliore la généralisation hors distribution. Enfin, NVIB améliore les performances hors distribution lors de l'adaptation fine sur des benchmarks de parole, texte, graphes et vision, confirmant son efficacité à induire des représentations généralisables dans divers modèles et tâches. Dans l'ensemble, cette thèse propose une perspective bayésienne variationnelle de l'attention, unifiant régularisation, explication et génération, et ouvrant de nouvelles voies pour faire progresser l'apprentissage de représentations.

Mots clefs : Transformers, Attention, Modèles à variables latentes, Inférence variationnelle, Méthodes bayésiennes non paramétriques, Processus de Dirichlet, Goulot d'étranglement informationnel, Apprentissage de représentations, Généralisation, Régularisation, Interprétabilité, Abstraction.

Zusammenfassung

Transformer-Modelle haben in verschiedenen Modalitäten – darunter Text, Graphen, Sprache und Bildverarbeitung – bemerkenswerte Erfolge erzielt, ermöglicht durch den Aufmerksamkeitsmechanismus. Dennoch sind die induktiven Vorannahmen, die bestimmen, wie Aufmerksamkeit Informationen enkodiert und Generalisierung unterstützt, noch nicht vollständig verstanden. Latente Variablenmodelle bieten einen fundierten Rahmen, um die enkodierte Information zu erklären, Generalisierung durch Regularisierung zu verbessern und generatives Modellieren zu ermöglichen. Die Anwendung solcher Modelle auf aufmerksamkeitbasierte Architekturen ist jedoch herausfordernd, da Aufmerksamkeit über Mengen operiert, die sowohl in ihrer Größe variabel als auch permutationsinvariant sind. Diese Dissertation führt das Nonparametric Variational Information Bottleneck (NVIB) ein – einen tiefen latenten Variablenrahmen, der Aufmerksamkeit als posteriori Inferenz über eine Dirichlet-Prozess-Mischung modelliert und sich damit auf natürliche Weise an diese mengenspezifischen Eigenschaften anpasst. Wir zeigen, dass NVIB das Training eines neuartigen, auf Transformern basierenden Variational Autoencoders von Grund auf ermöglicht, wobei die Anzahl der Embeddings reduziert und deren Inhalt gleichzeitig regularisiert wird. Als generatives Modell erlaubt NVIB eine glatte Interpolation und das Sampling in latenten Räumen variabler Größe. Bei Anwendung auf gestapelte Selbstaufmerksamkeits-Schichten induziert NVIB hierarchische Abstraktion, was die Interpretierbarkeit, Robustheit und sprachliche Ausrichtung verbessert. Dieses Rahmenwerk erlaubt zudem eine Reinterpretation vortrainierter Transformer als nichtparametrische variationale Modelle. NVIB zeigt auf, wie sie verlässliche von unzuverlässiger Information trennen und kodieren, was eine neuartige und kontrollierbare Regularisierung nach dem Training ermöglicht, die die Generalisierung außerhalb der Trainingsverteilung verbessert. Schließlich steigert NVIB die Leistung außerhalb der Trainingsverteilung beim Finetuning auf Sprach-, Text-, Graph- und Bild-Benchmarks und bestätigt so seine Wirksamkeit bei der Induktion generalisierbarer Repräsentationen über unterschiedliche Modelle und Aufgaben hinweg. Insgesamt bietet die Dissertation eine variational-bayessche Perspektive auf Aufmerksamkeit und vereint Regularisierung, Erklärung und Generierung, wodurch neue Wege für Fortschritte im Repräsentationslernen eröffnet werden.

Schlüsselwörter: Transformer, Aufmerksamkeit, Latente Variablenmodelle, Variationale Inferenz, Nichtparametrische bayessche Methoden, Dirichlet-Prozesse, Information Bottleneck, Repräsentationslernen, Generalisierung, Regularisie

Associated Publications

Thesis publications

The following chapters are based on research presented in these publications:

Chapter 3 is based on:

- "A VAE for Transformers with Nonparametric Variational Information Bottleneck",
  Henderson J., **Fehr F.**,
  ICLR 2023

Chapter 4 is based on:

- "Learning to Abstract with Nonparametric Variational Information Bottleneck"
  Behjati M.*, **Fehr F.***, Henderson J.,
  EMNLP 2023 Findings

Chapter 5 is based on:

- "Nonparametric Variational Regularisation of Pretrained Transformers",
  Fehr F., Henderson J.,
  COLM 2024

Chapter 6 is based on:

- "Fine-Tuning Pretrained Models with NVIB for Improved Generalisation"
  Fehr F., Baia A. E., Chang X., Coman A. C., El Hajal K., El Zein D., Kumar S., Zuluaga-Gomez J. P., Cavallaro A., Teney D., Henderson J.,
  ICLR 2025 Workshop on Spurious Correlation and Shortcut Learning

Other publications

Additional publications during the PhD include:

- "HyperMixer: An MLP-based Low Cost Alternative to Transformers",
  Mai F., Pannatier A., **Fehr F.**, Chen H., Marelli F., Fleuret F, Henderson J.,
  ACL 2023

- "CoRet: Improved Retriever for Code Editing",
  Fehr F., Sivaprasad P. T., Franceschi L., Zappella G.,
  ACL 2025

Contents

1 Motivation

1.1 Attention Mechanisms as Nonparametric Latent Variables

Modern machine learning is increasingly defined by its ability to generalise, generate, and interpret high-dimensional data. Nowhere is this more evident than in the widespread adoption of Transformer architectures. Transformers are remarkable for their capacity to learn generalisable representations across diverse domains. Pretraining followed by fine-tuning has become the de-facto paradigm, owing to its empirical success in transfer learning and scalability across language, vision, audio, and graph-based applications. Their success resides on a simple yet expressive mechanism: attention.

Attention enables interaction over permutation-invariant, variable-length sets of representations. This general mechanism supports scalability and integrates naturally with deep learning frameworks, enabling hierarchical computation through stacked layers. However, the nature of the latent representations learned in Transformers remains largely under-explored. Their ability to encode information, understand abstract concepts, and generalise beyond the training distribution suggests a rich internal structure. These representations often support strong performance with minimal supervision, yet their interpretability and inductive biases are not well understood.

Latent variable models offer a principled framework for understanding representation learning. By positing hidden structure underlying observed data, they provide a means to explain how representations encode and abstract information in low-dimensional spaces. When the underlying distribution is quantifiable, these models enable sampling, thereby supporting a generative interpretation. Through variational inference, they integrate naturally with deep networks and introduce stochasticity in a way that supports regularisation, robustness, and compression. These are essential properties for models that must generalise beyond their training distribution.

Latent variable models provide a natural lens through which to study the internal structure of attention-based architectures. They address key limitations in our understanding of Transformers by offering mechanisms to model abstraction, uncertainty, and compression. In particular, they support **regularisation**, by enforcing low-dimensional structure that improves generalisation; **explanation**, by revealing how attention mechanisms encode and abstract relevant information; and **generation**, by assessing whether representations capture the underlying data distribution in a coherent way. These facets, as shown in Figure 1.1, are central to the broader challenge of representation learning and motivate the latent variable perspective adopted throughout this thesis.

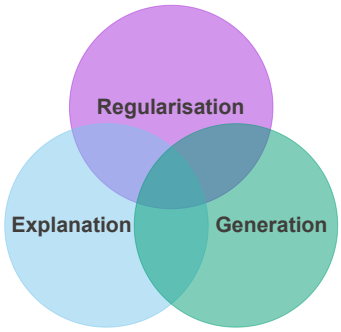


Figure 1.1: Regularisation, explanation, and generation form the core facets that motivate the latent variable perspective.

We turn to nonparametric distributions to model attention's permutation-invariant and unbounded set aggregation. Standard latent variable models assume a fixed latent dimension, but attention operates over sets of discrete tokens with variable size. These properties closely align with Dirichlet processes, which define infinite discrete mixtures that are both exchangeable and sparsity-inducing. The softmax aggregation in attention can be interpreted as producing a distribution over vectors, analogous to the mixture weights over components in a Dirichlet process. Table 1.1 summarises these parallels.

Table 1.1: Attention mechanisms and Dirichlet processes share core properties, motivating a nonparametric latent variable view of attention.

Attention Mechanisms	**Dirichlet Processes**
Unbounded representations	Infinite components
Permutation invariant	Exchangeable
$\sum_{i=1}^{n} \frac{\exp\left(\frac{1}{\sqrt{d}}\boldsymbol{u}\boldsymbol{z}_i^\top\right)}{\sum_{j=1}^{n}\exp\left(\frac{1}{\sqrt{d}}\boldsymbol{u}\boldsymbol{z}_j^\top\right)}\boldsymbol{z}_i$	$\sum_{i=1}^{\infty}\pi_i\delta_{\boldsymbol{z}_i}$

This connection motivates the core contribution of the thesis: a variational Bayesian framework that interprets attention as inference over a nonparametric mixture. We model the latent embeddings accessed by attention as latent variables drawn from an infinite mixture. This enables sparse information-theoretic regularisation, a distributional explanation of how information is encoded and abstracted, and a generative model of the input.

1.2 Research Questions & Contributions

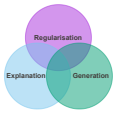
This section presents the core research questions and contributions of the thesis. Though the chapters follow a chronological engineering journey, the research targets fundamental challenges in representation learning. The resulting insights aim to generalise beyond specific models or implementations, lending the work longer-term relevance.

Research Question 1: ***How can we formulate deep variational latent variable models for attention-based architectures?***

We answer this question in Chapter 3 by developing a variational latent variable framework for attention-based architectures. We establish a theoretical connection between attention and Dirichlet processes, enabling a formulation of attention as a mixture distribution. This motivates the Nonparametric Variational Information Bottleneck (NVIB), implemented as a single-head cross-attention mechanism within a Transformer-based nonparametric variational autoencoder (NVAE). NVIB regularises the number of latent representations according to input complexity, which is a key property for handling variable-length text. Empirically, our NVAE retains core VAE behaviours such as reconstruction, generation, and smooth interpolation, while inducing embedding sparsity.

Research Question 2: ***How can hierarchical and interpretable abstraction of information be induced in Transformers?***

We answer this question in Chapter 4 by introducing a method for inducing interpretable abstractions in Transformers using NVIB. Applied across stacked self-attention layers, NVIB regularises the model by encouraging progressively sparser representations at deeper layers, allowing it to learn how many vectors are needed at each level. This leads to an emergent hierarchy of linguistic abstractions that often aligns with words or meaningful phrases, without relying on explicit supervision or tokenisation. Empirically, the model produces more interpretable and sparse attention patterns, captures stronger linguistic features, and demonstrates improved robustness.



Research Question 3: ***How do pretrained Transformer representations encode information and generalise out-of-distribution?***

We answer this question in Chapter 5 by extending NVIB to all forms of multi-head attention, establishing a connection to pretrained Transformers. This enables a reinterpretation of attention as variational inference, with identity initialisation and empirical priors providing a controllable post-training regularisation. Without additional training, this regularisation compresses redundant or unreliable information, leading to improved out-of-distribution generalisation on generative tasks such as summarisation and translation. These results offer a principled Bayesian explanation of how pretrained Transformers encode and generalise information.

Research Question 4: ***How can generalisable representations be induced during fine-tuning across different models and modalities?***

We answer this question in Chapter 6 by extending NVIB to regularise attention-based representations during fine-tuning across diverse models and modalities. We introduce engineering innovations to enable this, including a learnable prior for adaptability, clipped Dirichlet pseudo-counts for stability, and a simplified evaluation procedure. Applied to pretrained models in speech, text, graphs, and vision, NVIB improves out-of-distribution generalisation by regularising spurious features and encouraging robust, sparse representations. These results demonstrate that NVIB provides a unified and effective approach to inducing generalisable representations across modalities during fine-tuning.

2 Background

This chapter provides background on key concepts underpinning the thesis. We begin with Transformers and the mathematical formulation of attention, laying the groundwork for the technical contributions that follow. We then introduce deep latent variable models, the information bottleneck principle, and variational inference. Finally, we present nonparametric mixture distributions, whose properties form the probabilistic foundation of the proposed methods.

2.1 Transformers

Transformers (Vaswani et al., 2017) have become the foundation of modern sequence modelling, driving breakthroughs across language, vision, speech, and graphs. At their core lies the attention mechanism, a flexible, permutation-invariant function over variable-sized sets of vectors. This set-based formulation enables full parallelism during training, unlocking scalability, while naturally supporting key properties of language such as contextualisation and variable sequence length. We begin by describing the core building block, scaled dot-product attention. Next, we discuss multi-head attention, which enables multiple interactions between different representations. Finally, we present the complete Transformer architecture, which stacks multiple layers of attention and feed-forward networks.

2.1.1 Scaled Dot-Product Attention

The core operation underlying the Transformer is the *scaled dot-product attention* mechanism. Given a query matrix $\boldsymbol{Q} \in \mathbb{R}^{m \times d}$, key matrix $\boldsymbol{K} \in \mathbb{R}^{n \times d}$, and value matrix $\boldsymbol{V} \in \mathbb{R}^{n \times d}$, attention is computed as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{1}{\sqrt{d}}\boldsymbol{Q}\boldsymbol{K}^\top\right)\boldsymbol{V} \in \mathbb{R}^{m \times d} \tag{2.1}$$

Here, d denotes the dimensionality of the embedding space, while m and n represent the lengths of the query and key/value sequences, respectively. The dot product $\boldsymbol{Q}\boldsymbol{K}^\top$ measures the similarity between each query and all keys. The scaling factor $\frac{1}{\sqrt{d}}$ stabilises gradients during training by preventing large dot products. Assuming the query and key vectors are composed of independent random variables with zero mean and unit variance, their dot product has variance d and standard deviation $\sqrt{d}$. The softmax then

converts these similarities into attention weights, which are a probability distribution over keys. These weights are used to compute a weighted sum over the value vectors $\boldsymbol{V}$, producing an output that selectively aggregates relevant information from the input. This mechanism enables the model to flexibly combine context across the input sequence and stack such computations across layers.

The query, key, and value matrices are computed via learned linear projections of input embeddings. Specifically, the keys and values are obtained from a shared input $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$ using weight matrices $\boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{p \times d}$ and biases $\boldsymbol{b}^K, \boldsymbol{b}^V \in \mathbb{R}^d$. The queries are derived from a potentially different input $\boldsymbol{U}' \in \mathbb{R}^{m \times d}$ using projection parameters $\boldsymbol{W}^Q \in \mathbb{R}^{d \times d}$ and $\boldsymbol{b}^Q \in \mathbb{R}^d$:

$$\begin{aligned} \boldsymbol{K} &= \boldsymbol{Z}\boldsymbol{W}^K + \boldsymbol{b}^K, \\ \boldsymbol{V} &= \boldsymbol{Z}\boldsymbol{W}^V + \boldsymbol{b}^V, \\ \boldsymbol{Q} &= \boldsymbol{U}'\boldsymbol{W}^Q + \boldsymbol{b}^Q. \end{aligned}$$

In self-attention, the same input is used for all three projections ($\boldsymbol{U}' = \boldsymbol{Z}$ and $m = n$), while in cross-attention, the query input may differ from the key/value input. Causal attention further introduces a mask that blocks access to future positions, preserving the auto-regressive property necessary for generation.

This formulation allows attention to be expressed directly in terms of the transformer embedding spaces, with queries derived from $\boldsymbol{U}'$ and keys/values from $\boldsymbol{Z}$.

$$\text{Attention}(\boldsymbol{U}', \boldsymbol{Z}) = \text{Softmax}\left(\tfrac{1}{\sqrt{d}} \underbrace{(\boldsymbol{U}'\boldsymbol{W}^Q + \boldsymbol{b}^Q)}_{\boldsymbol{Q}} \underbrace{(\boldsymbol{Z}\boldsymbol{W}^K + \boldsymbol{b}^K)^\top}_{\boldsymbol{K}^\top}\right) \underbrace{(\boldsymbol{Z}\boldsymbol{W}^V + \boldsymbol{b}^V)}_{\boldsymbol{V}}$$

For notational convenience, we define the attention score matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ as the scaled dot-product interaction between the queries and keys:

$$\boldsymbol{A} = \tfrac{1}{\sqrt{d}} \underbrace{(\boldsymbol{U}'\boldsymbol{W}^Q + \boldsymbol{b}^Q)}_{\boldsymbol{Q}} \underbrace{(\boldsymbol{Z}\boldsymbol{W}^K + \boldsymbol{b}^K)^\top}_{\boldsymbol{K}^\top} \in \mathbb{R}^{m \times n}$$

Scaled dot-product attention captures only a single interaction between queries and keys. To model multiple relationships in parallel, transformers use *multi-head attention* with multiple independent projections.

2.1.2 Multi-Head Attention

Multi-Head Attention (MHA) extends scaled dot-product attention by allowing multiple interactions in parallel. The embedding dimension d is divided across h heads, each

learning its own attention pattern in a lower-dimensional subspace. The overall output is formed by concatenating the results from each head, as shown in Equation 2.2 and illustrated in Figure 2.1.

$$\text{MHA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}_{i=1}^{h} \left(\text{Softmax} \left(\frac{1}{\sqrt{d/h}} \boldsymbol{Q}_i \boldsymbol{K}_i^\top \right) \boldsymbol{V}_i \right) \tag{2.2}$$

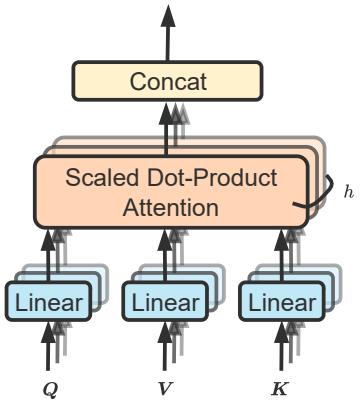


Figure 2.1: Multi-head attention: the queries, keys, and values are linearly projected and split across h heads. Each head performs scaled dot-product attention in parallel, and the results are concatenated to form the final output. Figure inspired by Vaswani et al. (2017)

.

In Multi-Head Attention, the query, key, and value projections are split across h parallel heads, each operating on a lower-dimensional subspace of size d/h. The projection weights are shaped accordingly: $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{h \times d \times \frac{d}{h}}$, with biases $\boldsymbol{b}^Q, \boldsymbol{b}^K, \boldsymbol{b}^V \in \mathbb{R}^{h \times \frac{d}{h}}$. This results in per-head query and key matrices of shape $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{h \times m \times \frac{d}{h}}$ and $\mathbb{R}^{h \times n \times \frac{d}{h}}$, respectively. Each head computes its own scaled dot-product attention. For head i, the attention scores $\boldsymbol{A}_i \in \mathbb{R}^{m \times n}$ are given by:

$$\boldsymbol{A}_i = \frac{1}{\sqrt{d/h}} \underbrace{(\boldsymbol{U'}\boldsymbol{W}_i^Q + \boldsymbol{b}_i^Q)}_{\boldsymbol{Q}_i} \underbrace{(\boldsymbol{Z}\boldsymbol{W}_i^K + \boldsymbol{b}_i^K)^\top}_{\boldsymbol{K}_i^\top} \in \mathbb{R}^{m \times n} \tag{2.3}$$

For completeness we can write multi-head attention as a function of the original embedding space of transformers $\boldsymbol{U}'$ and $\boldsymbol{Z}$.

$$\text{MHA}(\boldsymbol{U}', \boldsymbol{Z}) = \text{Concat}_{i=1}^{h} \left(\text{Softmax} \left(\frac{1}{\sqrt{d/h}} \underbrace{(\boldsymbol{U'}\boldsymbol{W}_i^Q + \boldsymbol{b}_i^Q)}_{\boldsymbol{Q}_i} \underbrace{(\boldsymbol{Z}\boldsymbol{W}_i^K + \boldsymbol{b}_i^K)^\top}_{\boldsymbol{K}_i^\top} \right) \underbrace{(\boldsymbol{Z}\boldsymbol{W}_i^V + \boldsymbol{b}_i^V)}_{\boldsymbol{V}_i} \right)$$

In the next section, we consider the full Transformer architecture, which integrates attention with feed-forward layers, residual connections, and normalisation to enable deep sequence modelling.

2.1.3 Transformer Architecture

The Transformer architecture (Figure 2.2) processes sequences of tokens by combining multi-head attention with feedforward neural networks. Residual connections and layer normalisation (add & norm) ensure stable and efficient training when stacking multiple layers. Attention mechanisms model relationships between tokens, capturing contextual dependencies, while feed-forward layers capture and process deep features within each token's representation.

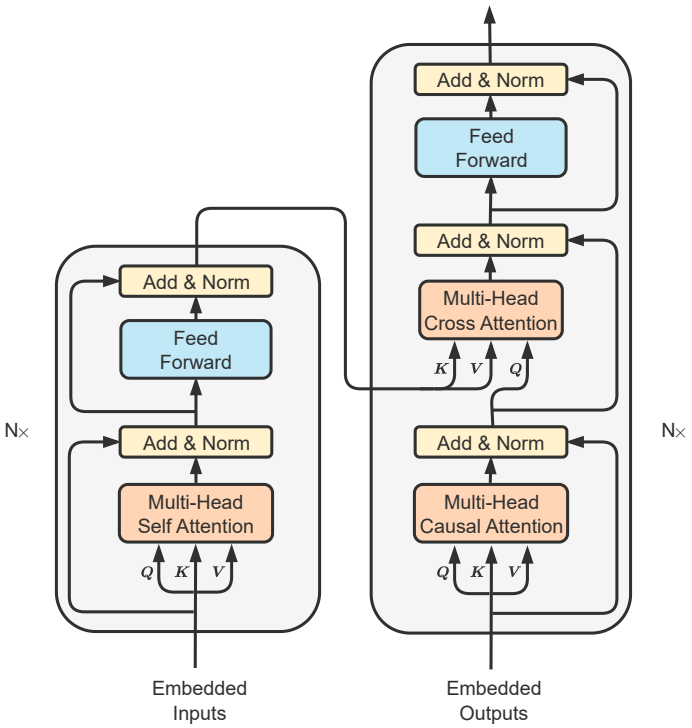


Figure 2.2: The Transformer architecture: encoder (left) and decoder (right). Figure inspired by Vaswani et al. (2017)

.

The Transformer treats tokens as a set. Positional information is incorporated through learned or fixed positional embeddings, allowing the model to capture token order directly within the features.

The encoder (left) transforms an input sequence into a set of contextualised embeddings. Each token's representation is informed by its relationship to others in the sequence, captured through self-attention. This is followed by a position-wise feedforward network (typically a multi-layer perceptron) applied independently to each token. Together, these components integrate contextual information with non-linear feature transformations. Encoders are commonly used for tasks such as classification.

The decoder (right) generates an output sequence conditioned on both previously generated tokens and the encoder's output. Each decoder layer begins with a self-attention mechanism, constrained by causal masking to ensure that each position attends only to earlier tokens, preserving the autoregressive property. This is followed, optionally, by a cross-attention mechanism, which allows the decoder to incorporate contextual information from the encoder's embeddings. As with the encoder, a position-wise feed-forward network is then applied independently to each token. During training, teacher forcing is used, feeding ground truth tokens to guide learning. At inference, generation proceeds autoregressively. This architecture underpins many large language models (LLMs), enabling coherent and fluent sequence generation.

The full Transformer architecture combines an encoder and decoder to support flexible and expressive sequence-to-sequence modelling. It is widely used in tasks such as machine translation, summarisation, and question answering, where the encoder produces contextual embeddings and the decoder generates the output sequence. This model serves as the foundation for the rest of the thesis.

2.2 Variational Latent Variable Models

In this section, we view latent variable models through an information-theoretic lens. The information bottleneck principle is used as a regularisation method to reduce generalisation error. We then introduce Variational Information Bottleneck (VIB), which adapts this idea for deep models.

2.2.1 Information Bottleneck Principle

The Information Bottleneck (IB) principle (Tishby et al., 2000) offers a framework for learning optimally compressed representations for a specific task and control generalisation error (Kawaguchi et al., 2023). At its core, the IB principle seeks to balance two competing objectives: compression, where the learned latent representation should be as succinct as possible, removing irrelevant information from the input; and predictive power, where the representation must retain as much information about the target variable to perform the downstream task.

Consider an input random variable X (e.g., a sentence or image) and a target random variable Y (e.g., a class label). Our goal is to find a compressed latent variable Z from X. This latent variable Z should be maximally informative about Y whilst being minimally informative about X, beyond what is necessary to predict Y. In essence, Z acts as a "bottleneck" through which information flows from X to Y.

Mathematically, the IB principle formalises this trade-off using mutual information. The objective function to optimise for the latent variable Z is given by:

$$\mathcal{L}_{\text{IB}} = \underbrace{I(Z;Y)}_{\text{Predict}} - \beta \underbrace{I(Z;X)}_{\text{Compress}} \tag{2.4}$$

Here, $I(Z;Y)$ quantifies the relevance of the latent variable Z to the target Y; maximising this term encourages Z to capture as much information about Y as possible. Conversely, $I(Z;X)$ quantifies the redundancy of Z with respect to the input X. Minimising this term promotes a compressed representation, forcing Z to discard information from X that is not relevant to Y. The term $\beta \geq 0$ is a hyperparameter that controls the balance between these compression and predictive power objectives. A higher value of β increases compression, whilst a lower β prioritises retaining more information about Y.

Directly optimising the IB objective function $\mathcal{L}_{\text{IB}}$ is often intractable for complex, high-dimensional data, due to the computational demands of the mutual information terms. This limitation motivates the need for a tractable approximation, which leads us to the Variational Information Bottleneck (VIB) framework.

2.2.2 Variational Information Bottleneck

The Information Bottleneck principle offers a compelling theoretical framework for learning optimal representations. However, its practical application is often limited by the intractability of computing the true posterior distribution $p(Z|X)$ or the marginal likelihood $p(X)$. This intractability arises with large datasets or complex models. As a result, sampling-based inference methods such as Markov Chain Monte Carlo (MCMC) or Gibbs sampling can become prohibitively slow.

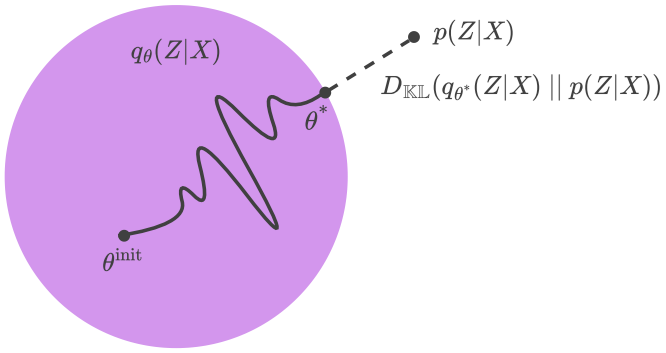


Figure 2.3: Variational Inference optimises the parameters θ of the approximating distribution $q_\theta(Z|X)$ to match the true posterior $p(Z|X)$ by minimising the Kullback–Leibler divergence. Figure inspired by Blei (2017).

Variational Inference Rather than rely on slow sampling methods, Variational Inference (VI) turns inference into an optimisation problem. It approximates the intractable posterior $p(Z|X)$ with a simpler, tractable distribution $q_\theta(Z|X)$, where θ parameterises the distribution. This surrogate, known as the variational distribution, is chosen from a family that allows efficient computation. The goal is to make $q_\theta(Z|X)$ closely match $p(Z|X)$ by minimising their Kullback–Leibler (KL) divergence (Figure 2.3).

Evidence Lower Bound (ELBO) The "closeness" between $q_\theta(Z|X)$ and $p(Z|X)$ is typically quantified using the Kullback-Leibler (KL) divergence, $\mathrm{D}_{\mathbb{KL}}(q(_\theta Z|X) \,||\, p(Z|X))$. Our goal in variational inference is to minimise this KL divergence by optimising θ. Minimising $\mathrm{D}_{\mathbb{KL}}(q_\theta(Z|X) \,||\, p(Z|X))$ corresponds to maximising a quantity known as the Evidence Lower Bound (ELBO).

Theorem 2.2.1. *Let $q(Z|X)$ be the distribution over latent variables Z given observed data X, and let $p(Z|X)$ be the true posterior under the model joint $p(X, Z)$. Then, the log marginal likelihood $\log p(X)$ has the following lower bound:*

$$\log p(X) \geq \mathbb{E}_{q(Z|X)}[\log p(X|Z)] - \mathrm{D}_{\mathbb{KL}}(q(Z|X) \,||\, p(Z))$$

*This bound is called the **Evidence Lower Bound (ELBO)**.*

Proof.

$$\begin{aligned}
\log p(X) &= \log \int p(X, Z)\, dZ \\
&= \log \int \frac{p(X, Z)}{q(Z|X)} q(Z|X)\, dZ && \text{(Multiply and divide by } q(Z|X)) \\
&= \log \mathbb{E}_{q(Z|X)} \left[\frac{p(X, Z)}{q(Z|X)} \right] && \text{(Expectation over } q(Z|X)) \\
&\geq \mathbb{E}_{q(Z|X)} \left[\log \frac{p(X, Z)}{q(Z|X)} \right] && \text{(Jensen's inequality: } \log \mathbb{E}(.) \geq \mathbb{E}[\log .]) \\
&= \mathbb{E}_{q(Z|X)} \left[\log p(X|Z) + \log p(Z) - \log q(Z|X) \right] && \text{(Factor joint: } p(X, Z) = p(X|Z)p(Z)) \\
&= \mathbb{E}_{q(Z|X)}[\log p(X|Z)] - \mathrm{D}_{\mathbb{KL}}(q(Z|X) \,||\, p(Z)) && \text{(Group likelihood and KL divergence)}
\end{aligned}$$

□

The ELBO, $\mathcal{L}_{\text{ELBO}}$, thus serves as a lower bound on the log marginal likelihood. Maximising the ELBO amounts to minimising $\mathrm{D}_{\mathbb{KL}}(q(Z|X)||p(Z|X))$, thereby pushing the variational distribution $q(Z|X)$ closer to the true posterior $p(Z|X)$.

In the context of the Information Bottleneck, the VIB framework maps the IB objective $\mathcal{L}_{\text{IB}}(Z) = I(Z;Y) - \beta I(Z;X)$ to a variational objective that can be optimised. Specifically, $I(Z;Y)$ is lower-bounded by $\mathbb{E}_{q(Z|X)}(\log p(Y|Z))$, which contributes to the predictive power. The compression term, $I(Z;X)$, is approximated by $\mathrm{D}_{\mathbb{KL}}(q(Z|X)||p(Z))$, where

$p(Z)$ is a simple prior distribution. This leads to the VIB objective function, a generalisation of the ELBO tailored for information-theoretic compression:

$$\mathcal{L}_{\text{VIB}} = \underbrace{\mathbb{E}_{q_\theta(Z|X)}(\log p_\phi(Y|Z))}_{\text{predict}} - \beta \underbrace{\mathrm{D}_{\mathbb{KL}}(q_\theta(Z|X)||p(Z))}_{\text{compress}}$$

Here, the distributions $q_\theta(Z|X)$ and $p_\phi(Y|Z)$ are parameterised by θ and ϕ, respectively, and jointly optimised.

Reparameterisation Trick In deep latent-variable models, sampling introduces a challenge, as the gradients cannot pass through the stochastic sampling step. This disrupts the gradient flow required for optimisation. The *reparameterisation trick* addresses this by rewriting the sampling process as a deterministic function of the distribution parameters and an auxiliary noise variable.

For instance, rather than sampling $z \sim \mathcal{N}(\mu, \sigma^2)$, or equivalently $z \sim p(z \mid \mu, \sigma)$, with p denoting a Gaussian conditioned on its parameters—we instead sample $\epsilon \sim \mathcal{N}(0, 1)$ and compute $z = \mu + \sigma\,\epsilon$. This transformation, known as a *location-scale reparameterisation*, preserves differentiability with respect to μ and σ. Figure 2.4 illustrates this idea.

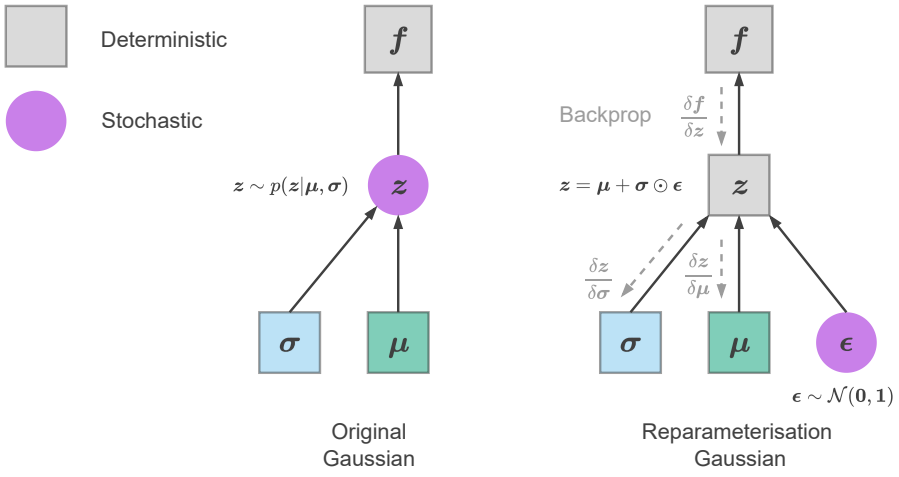


Figure 2.4: Illustration of the reparameterisation trick for a Gaussian latent variable. **Left:** Original stochastic sampling process. **Right:** Equivalent deterministic computation using a standard Gaussian noise variable and distribution parameters, enabling gradient backpropagation through the sample.

When this is not feasible, inverse cumulative distribution function (CDF) sampling may be used, provided the inverse exists in closed form. For more complex cases, such as Gamma or Dirichlet distributions, one must rely on approximations or implicit reparameterisation gradients to maintain a differentiable computation graph during training.

2.3 Nonparametric Mixture Distributions

In this section, we largely follow the notation of Teh (2010) and Frigyik et al. (2010).

A fundamental challenge in statistical modelling is to infer an underlying, unknown distribution, F, from observed data. Standard parametric approaches assume F belongs to a pre-specified family of distributions, defined by a finite number of parameters (e.g., a Gaussian distribution characterised by its mean and variance). This assumption, however, can lead to model misspecification and underfitting if the true data-generating process deviates significantly from the chosen parametric form. Bayesian nonparametric models offer an alternative by allowing for models of unbounded complexity. This inherently mitigates underfitting. However, overfitting is naturally addressed within the Bayesian framework through approximation (e.g., via variational methods) of the posterior distribution.

Consider the task of inferring F given independent and identically distributed observations $x_1, \ldots, x_n \sim F$. A Bayesian approach necessitates placing a prior distribution over F to subsequently compute the posterior $P(F \mid x_1, \ldots, x_n)$. Crucially, for the nonparametric approach to be practical, posterior computations must remain tractable, even when the prior is defined over an infinite-dimensional space of distributions.

2.3.1 Multi-Definitions of Dirichlet Processes

The Dirichlet Process (DP) is a stochastic process used in Bayesian nonparametric models, notably within Dirichlet Process Mixture Models (often termed infinite mixture models). It is a distribution over distributions, meaning that each draw from a DP is itself a random discrete distribution. The name derives from its property of having Dirichlet-distributed finite-dimensional marginals. Conceptually, the DP generalises the Dirichlet distribution to infinite-dimensional settings, extending its support to spaces such as vector spaces.

The utility of DPs stems from several key properties. Primarily, a DP provides a prior over discrete distributions. These distributions are nonparametric, adapting their complexity to observed data without a fixed number of parameters. DPs also offer analytically tractable posterior distributions due to their conjugate prior property, greatly simplifying Bayesian inference by often preserving the DP form with updated parameters. Furthermore, the DP addresses the challenging problem of pre-specifying mixture components. Its flexibility allows the number of effective components to grow naturally with the data, supporting a theoretically unbounded number and enabling modelling arbitrary complexity. Finally, observations from a Dirichlet Process mixture are exchangeable, a fundamental property implying that observation order does not affect joint probability, allowing for an infinite sequence of data points. Finally, observations from a Dirichlet Process mixture are exchangeable, meaning their joint distribution is invariant to ordering. This property

underpins the use of DPs for modelling sequences of arbitrary length and is formally justified by De Finetti's theorem, discussed in Section 2.3.3.

Definition via Finite Dirichlet Marginals The Dirichlet Process (DP) is a random distribution F on a space (Θ) with base distribution G_0 and concentration parameter $\alpha > 0$, denoted $F \sim \text{DP}(\alpha, G_0)$. Its original definition states that for any finite measurable partition $A_1, \ldots, A_r$ of Θ, the vector $(F(A_1), \ldots, F(A_r))$ follows a Dirichlet distribution:

$$(F(A_1),\ \ldots,\ F(A_r)) \sim \text{Dir}(\alpha G_0(A_1),\ \ldots,\ \alpha G_0(A_r)) \tag{2.5}$$

The base distribution G_0 acts as the mean of the DP $\mathbb{E}[F(A)] = G_0(A)$, while the concentration parameter α controls its variability around G_0, behaving as an inverse variance $\text{Var}[F(A)] = G_0(A)(1 - G_0(A))/(\alpha + 1)$. As $\alpha \to \infty$, the lower the variance and F converges to G_0. A key property is that draws from a DP are discrete distributions with probability one, even if G_0 is continuous.

Definition as Infinite Dirichlet Distributions A useful definition of a Dirichlet Process, $F \sim \text{DP}(\alpha_0, G_0)$, is as the limit of a sequence of finite, symmetric Dirichlet distributions (Teh, 2010), expressed as an infinite sum of Dirac delta functions:

$$\begin{aligned} F &= \sum_{i=1}^{\infty} \pi_i \delta_{\boldsymbol{z}_i} \\ \boldsymbol{\pi} &\sim \lim_{\kappa \to \infty} \text{Dir}(\frac{\alpha_0}{\kappa},\ \ldots^{\kappa},\ \frac{\alpha_0}{\kappa}) \\ \boldsymbol{z}_i &\sim G_0 \quad \text{for } i = 1,\ \ldots,\ \infty \end{aligned} \tag{2.6}$$

Here, $\boldsymbol{z}_i$ are independently drawn from G_0, and the weights $\boldsymbol{\pi}$ are derived from a symmetric Dirichlet distribution in the infinite limit. Despite the symmetric construction, the resulting weights π_i are highly asymmetric, with a few categories receiving a large proportion of mass and an infinitely long tail of infinitesimally probable categories.

Alternative Definitions Beyond these definitions, the Dirichlet Process can also be defined or sampled using several constructive approaches. The Stick-Breaking Construction generates the infinite sequence of weights (π_i) by successively breaking off proportions from a unit-length "stick", and associating these with sample drawn from G_0. The Chinese Restaurant Process (CRP) provides a sequential probabilistic model for partitioning data into clusters. The Blackwell-MacQueen Urn Scheme is another sequential scheme for sampling from the predictive distribution of a DP. Related processes include the Indian Buffet Process (IBP), used for modelling features, and the more general Pitman-Yor Process (PYP), which offers a richer-gets-richer behaviour.

2.3.2 Dirichlet Process Posterior

A key property of the Dirichlet Process is its conjugacy. If the prior over a random distribution F is a Dirichlet Process, then the posterior distribution of F is also a Dirichlet Process. This tractability is crucial for performing Bayesian inference with DP models.

Specifically, if we have a prior $F \sim \text{DP}(\alpha_0, G_0)$ and observe n independent and identically distributed draws $\theta_1, \ldots, \theta_n$ from F, the posterior distribution of F can be derived by examining its finite-dimensional marginals. For any finite measurable partition $A_1, \ldots, A_r$ of Θ, the posterior marginals are given by:

$$(F(A_1), \ldots, F(A_r)) \mid \theta_1, \ldots, \theta_n \sim \text{Dir}(\alpha_0 G_0(A_1) + n_1, \ldots, \alpha_0 G_0(A_r) + n_r) \tag{2.7}$$

where n_j is the count of observations falling into partition A_j. Since this holds for all finite measurable partitions, the posterior distribution over F must itself be a Dirichlet Process (Teh, 2010):

$$F \mid \theta_1, \ldots, \theta_n \sim \text{DP}\left(\alpha_0 + n, \frac{\alpha_0 G_0 + \sum_{i=1}^{n} \delta_{\theta_i}}{\alpha_0 + n}\right) \tag{2.8}$$

Here, δ_{θ_i} denotes a Dirac distribution at point θ_i. The posterior Dirichlet Process has an updated concentration parameter, which is the sum of the prior concentration parameter and the number of observations ($\alpha_0 + n$). The posterior base distribution is a weighted average of the prior base distribution G_0 and the empirical distribution of the observed data $\sum_{i=1}^{n} \delta_{\theta_i}$. This update mechanism demonstrates how the posterior distribution shifts its mass towards the observed data while still retaining influence from the prior.

In the context of Dirichlet Process Mixture Models, the posterior parameters (denoted with superscript q, with prior parameters having superscript p) are often expressed as:

$$F \sim \text{DP}\left(G_0^q,\ \alpha_0^q\right) \tag{2.9}$$

$$\alpha_0^q = \sum_{i=1}^{n+1} \alpha_i^q$$

$$G_0^q = \sum_{i=1}^{n+1} \frac{\alpha_i^q}{\alpha_0^q} G_i^q$$

Here, $n+1$ represents the number of components in this specific posterior representation, where the $(n+1)^{\text{th}}$ component stems from the prior, with $\alpha_{n+1}^q = \alpha_0^p$ and $G_{n+1}^q = G_0^p$. Each α_i^q and G_i^q for $i = 1, \ldots, n$ typically correspond to the posterior parameters associated with an observed data point θ_i, while G_0^q is their weighted mixture and α_0^q is the total concentration.

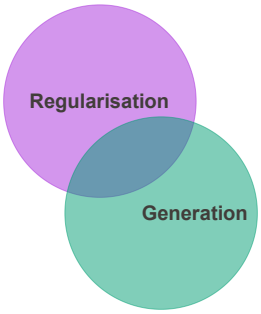
2.3.3 Properties of Dirichlet Processes

Exchangeability The exchangeability property of observations drawn from a Dirichlet Process is directly linked to de Finetti's theorem (Teh, 2010). De Finetti's theorem states that an exchangeable sequence of random variables is conditionally independent given some latent variable. In the context of a Dirichlet Process, this latent variable is the random distribution F. Therefore, for a sequence of observations $x_1, x_2, \ldots$ drawn from $F \sim \text{DP}(\alpha, G_0)$, these observations are exchangeable and conditionally independent given F.

Aggregation Property The Dirichlet distribution exhibits an intuitive aggregation property (Frigyik et al., 2010, Section 1.3). This property states that if a vector of probabilities $\mathbf{Q} = (Q_1, \ldots, Q_k)$ is Dirichlet distributed with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$, then for any finite measurable partition $A_1, \ldots, A_r$ of the indices $\{1, 2, \ldots, k\}$, the aggregated sums of probabilities $(\sum_{i \in A_1} Q_i, \ldots, \sum_{i \in A_r} Q_i) \sim \text{Dir}(\sum_{i \in A_1} \alpha_i, \ldots, \sum_{i \in A_r} \alpha_i)$ are also Dirichlet distributed with aggregated parameters. This means that if parts of the sample space are merged together, the resulting probabilities over these new aggregated events still follow a Dirichlet distribution with corresponding summed parameters.

Neutrality A random vector $\mathbf{Q} = (Q_1, \ldots, Q_k)$ is said to be neutral if, for each component j, Q_j is independent of the random vector formed by scaling the remaining components by $1/(1 - Q_j)$ (Frigyik et al., 2010, Section 2.2.2). Formally, $\mathbf{Q}$ is neutral if for each $j \in \{1, \ldots, k\}$, Q_j is independent of $\frac{1}{1-Q_j}(Q_1, \ldots, Q_{j-1}, Q_{j+1}, \ldots, Q_k)$. Dirichlet distributions possess this property of neutrality, meaning that removing and re-normalising a subset of components does not affect the distribution of the remaining scaled components.

3 A VAE for Transformers with NVIB



Research Question

How can we formulate deep variational latent variable models for attention-based architectures?

Chapter Summary

We address the formulation of deep variational latent variable models for attention-based architectures by developing a Variational AutoEncoder (VAE) for Transformers. We introduce a Variational Information Bottleneck (VIB) framework for attention-based embeddings, which we formalise as mixture distributions. Leveraging Bayesian nonparametrics, we develop a Nonparametric VIB (NVIB) tailored to these representations. The variable number of mixture components supported by nonparametrics captures the variable number of vectors supported by attention, and exchangeable distributions from nonparametrics capture the permutation invariance of attention. Our Transformer VAE (NVAE) uses NVIB to regularise the information passing from the Transformer encoder to the Transformer decoder. Evaluations of a NVAE, trained on natural language text, demonstrate that NVIB can regularise the number of mixture components in the induced embedding whilst maintaining generation quality and reconstruction capacity.

Publication

- "A VAE for Transformers with Nonparametric Variational Information Bottleneck", Henderson J., **Fehr F.**, *ICLR 2023*

Code Repository

The code is publically available at:

- https://github.com/idiap/nvib
- https://github.com/idiap/nvib transformers.

Author Contributions

James Henderson contributed the high-level vision and theoretical derivations on Bayesian nonparametrics. Fabio Fehr was responsible for the implementation and experiment design, working in a feedback loop with the derivation to ensure practicality. He contributed to the reparametrisation tricks and factorised DP sampling, which enabled the final evaluations.

3.1 Modelling Attention with Variational Latent Variables

Attention-based deep learning models, such as Transformers (Vaswani et al., 2017; Devlin et al., 2019), have achieved unprecedented empirical success in a wide range of cognitive tasks, in particular in natural language processing. The use of attention allows these models to represent their input with multiple vectors, which is essential for embedding natural language text (Bahdanau et al., 2015). On the other hand, deep variational latent variable approaches to representation learning, such as variational autoencoders (VAEs) (Kingma and Welling, 2014), have also been shown to have many benefits (Mathieu et al., 2019; Ghosh et al., 2020; Vahdat and Kautz, 2020), especially due to their variational information bottleneck (VIB) (Alemi et al., 2017) for regularising the induced latent representations. However, it has not been clear how to combine these two trends, because the latent space induced by Transformers is a set of vectors whose size grows with the size of the input, whereas standard VIB methods only apply to a vector space of a fixed size (Liu and Liu, 2019; Fang et al., 2021; Park and Lee, 2021). To define a VIB regulariser for a Transformer's embedding space, we need to allow the size of a latent representation to vary dynamically depending on the complexity of the individual input, and yet regularise the total amount of information conveyed by the whole representation.

In this chapter, we propose such a variational information bottleneck for variable sized latent representations, which we use to regularise the embeddings of a Transformer encoder-decoder, giving us a variational autoencoder for Transformers. Like a Transformer encoder's embedding space, the proposed VAE's sampled encoder output is (a generalisation of) a set of vectors, and the decoder accesses this embedding with (a generalisation of) cross attention. But unlike Transformers, the proposed VIB layer for this VAE regularises the (effective) number of vectors in the set, as well as the information conveyed by each vector. We show that this regularisation improves generative abilities and compresses latent representations. In addition to the regularisation of overparameterised language models (Child et al., 2019), previous work shows the efficacy of VAEs for: disentanglement (Higgins et al., 2017), language generation (Liu and Liu, 2019), and explainability (Mercatali and Freitas, 2021). All these topics are important and active areas of research in NLP.

To define this VIB, we need to model distributions over these variable-sized encoder embeddings, as interpreted by cross attention. Firstly, because the attention function returns an interpolation between the vectors output by the encoder, it generalises across the varying number of vectors, which like the input length is theoretically unbounded. Thus, to define distributions over these unbounded embeddings, we need to use nonparametric methods (Jordan, 2010). Secondly, the attention function is insensitive to the order of the vectors output by the encoder, so it interprets this embedding as a permutation-invariant set of vectors. Thus, the distributions over these permutation-invariant embeddings should be exchangeable (Jordan, 2010). Thirdly, the attention function imposes a normalised weighting over the embedding vectors, via the attention weights. So we should model an embedding as a distribution rather than a set.

A normalised weighting over an unbounded permutation-invariant set of fixed-length vectors matches exactly the properties of a nonparametric space of mixture distributions, which have been extensively studied in Bayesian nonparametrics using exchangeable distributions (Blei and Jordan, 2006; Jordan, 2010). In previous work, Bayesian nonparametrics is typically applied to learning *models* where the number of parameters grows with the size of the *training data* (Teh, 2010; Jordan, 2010; Kossen et al., 2021). In contrast, we apply it to inferring *latent representations* where the number of parameters grows with the size of the *input*. We believe this is the first work to use nonparametric methods in this way for deep variational latent variable models.

To define a precise equivalence between attention-based representations and mixture distributions, we provide an interpretation of attention where the input set of vectors defines a mixture of impulse distributions, which is used as a prior to denoise the observed query vector. Generalising sets of vectors to mixture distributions and generalising the attention function to query denoising allows us to propose a general deep variational Bayesian framework for attention-based models using Bayesian nonparametrics. More specifically, we propose to use Dirichlet processes (DPs) as the exchangeable distributions (Aldous, 1985; Jordan, 2010) to specify distributions over mixtures of impulse distributions, including distributions over the effective number of components in the mixture.

We define a nonparametric VIB (NVIB) layer using a DP prior and posterior to regularise the effective size of variable-sized latent representations. This NVIB layer uses exact inference to infer the posterior from a set of pseudo-observations, and uses proposed efficient approximations to sample from this posterior with a reparameterisation trick and to regularise it with the KL divergence with the prior. Applying this NVIB regulariser to a Transformer autoencoder gives us our proposed nonparametric variational autoencoder (NVAE). The noise introduced by sampling from the DP posterior controls the amount of information which flows from the encoder to the decoder, despite the fact that the amount of information required to reconstruct different text inputs varies enormously.

To evaluate the effectiveness of NVIB, we train a NVAE on natural language text and find that it is able to reconstruct, generate and regularise the effective number of vectors in the latent representation, thereby demonstrating that NVAE is a viable VAE. We also find that the regularised latent space is smooth, using a proposed method for interpolating between DP posteriors to generate interpolations between sentences.

Contributions This chapter makes the following contributions: (1) We propose a variational Bayesian framework for modelling attention-based representations using mixture distributions, denoising attention and Bayesian nonparametrics (Section 3.2). (2) We propose a nonparametric variational information bottleneck (NVIB) regulariser for learning attention-based representations (Section 3.3). (3) We propose a nonparametric varia-

tional autoencoder (NVAE), which is a variational Bayesian extension of a Transformer encoder-decoder (Section 3.4). (4) We show that the NVAE model is a competitive VAE which can reconstruct, generate, regularise its latent space and intuitively interpolate between sentences (Section 3.5).

Related work Related work in stochastic attention assume that the keys, queries, values (Martin et al., 2020) or attention weight vectors of the network are treated as latent random variables (Deng et al., 2018; Bahuleyan et al., 2018; Fan et al., 2020; Cinquin et al., 2022). Nguyen et al. (2022) provides a formulation and interpretation of attention keys as latent mixture distributions, whereas our formulation characterises the whole attention function and is interpreted as Bayesian query denoising. The use of Bayesian nonparametrics to learn a variable sized latent space using a VAE (Nalisnick and Smyth, 2017; Goyal et al., 2017; Echraibi et al., 2020) still assumes a fixed-sized latent representation at test time, unlike our proposal.

3.2 Nonparametric Bayesian Transformer Embeddings

This section proposes a formalisation of attention-based embeddings as mixture distributions over a vector space, and proposes nonparametric Bayesian methods for modelling information about these mixture distributions. First we show that standard attention functions can be interpreted as implementing Bayesian query denoising, where the set of vectors being accessed specifies a mixture of impulse distributions (Section 3.2.1). We adopt mixture distributions over vectors as a generalisation of attention-based representations, and adopt this denoising function as a generalisation of attention. Then we use Bayesian nonparametrics to propose prior (Section 3.2.2) and posterior distributions (Section 3.2.3) over these mixture distributions. These priors and posteriors form the basis of our nonparametric variational information bottleneck, proposed in Section 3.3.

3.2.1 Denoising Attention

In this section, we show that the attention function can be defined as Bayesian posterior inference, a perspective we refer to as *Bayesian query denoising*. We call this general interpretation of attention *Denoising Attention*. As the basis of our approach to attention-based representations, we generalise the set of vectors to a probability distribution over vectors, and generalise attention to a function of these probability distributions.

The attention mechanism we assume is scaled dot product attention, standardly used in many attention-based models, including Transformers. For simplicity, we consider single-head cross attention, where a single query vector is mapped to a single result vector and ignore bias terms in the linear projections. Later we will generalise for multi query and multi-head attention.

This attention function uses keys $\boldsymbol{K} \in \mathbb{R}^{n\times d}$ and values $\boldsymbol{V} \in \mathbb{R}^{n\times d}$ which are both projections from the same set of vectors $\boldsymbol{Z} \in \mathbb{R}^{n\times p}$ via weight matrices $\boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{p\times d}$. The query $\boldsymbol{q} \in \mathbb{R}^{1\times d}$ is a projection from the input $\boldsymbol{u'} \in \mathbb{R}^{1\times p}$ via the weight matrix $\boldsymbol{W}^Q \in \mathbb{R}^{p\times d}$. The keys' dimensionality d is used for scaling. Scaled dot product attention is then defined as:

$$\begin{aligned}
\text{Attention}(\boldsymbol{u}', \boldsymbol{Z}) &= \text{softmax}\left(\tfrac{1}{\sqrt{d}}\underbrace{(\boldsymbol{u'W}^Q)}_{\boldsymbol{Q}}\underbrace{(\boldsymbol{ZW}^K)^\top}_{\boldsymbol{K}^\top}\right)\underbrace{\boldsymbol{ZW}^V}_{\boldsymbol{V}} \\
&= \text{softmax}\left(\tfrac{1}{\sqrt{d}}\underbrace{\boldsymbol{u'W}^Q(\boldsymbol{W}^K)^\top}_{\boldsymbol{u}}\boldsymbol{Z}^\top\right)\boldsymbol{ZW}^V \\
&= \text{softmax}\left(\tfrac{1}{\sqrt{d}}\boldsymbol{u}\boldsymbol{Z}^\top\right)\boldsymbol{ZW}^V \\
&= \text{Attn}(\boldsymbol{u}, \boldsymbol{Z})\boldsymbol{W}^V
\end{aligned}$$

where $\boldsymbol{u} = \boldsymbol{u'W}^Q(\boldsymbol{W}^K)^\top \in \mathbb{R}^{1\times p}$ is a projected query $\boldsymbol{u'}$ into the space of $\boldsymbol{Z}$. In the last line we rewrite scaled dot product attention in terms of a core dot product attention function $\text{Attn}(\boldsymbol{u}, \boldsymbol{Z})$ where all operations are done in the space of $\boldsymbol{Z}$:

$$\begin{aligned}
\text{Attn}(\boldsymbol{u}, \boldsymbol{Z}) &= \text{softmax}\left(\tfrac{1}{\sqrt{d}}\boldsymbol{u}\boldsymbol{Z}^\top\right)\boldsymbol{Z} \\
&= \sum_{i=1}^{n}\frac{\exp(\frac{1}{\sqrt{d}}\boldsymbol{u}\boldsymbol{z}_i^\top)}{\sum_{i=1}^{n}\exp(\frac{1}{\sqrt{d}}\boldsymbol{u}\boldsymbol{z}_i^\top)}\boldsymbol{z}_i
\end{aligned}$$

This formulation allows us to interpret attention as accessing a set of vectors in latent space of Transformers.

Attention as Bayesian Query Denoising We show that the attention function $\text{Attn}(\boldsymbol{u}, \boldsymbol{Z})$ can be expressed in two equivalent ways: first, as the standard weighted sum over the vectors $\boldsymbol{z}_i \in \boldsymbol{Z}$; and second, as an expectation with respect to a discrete probability distribution supported on the same $\boldsymbol{z}_i$. To build intuition, Figure 3.1 shows the attention mechanism in 2D. The output is an interpolation between the key vectors $\boldsymbol{Z}$ which are close to the query, determined by their softmax scores. Figure 3.2 generalises this idea and reinterprets attention as an expectation. The keys $\boldsymbol{Z}$ are treated as point masses (impulses), forming a discrete probability distribution. The query $\boldsymbol{u}$ is viewed as a noisy observation of some latent true vector $\boldsymbol{v}$, drawn from the distribution defined by $\boldsymbol{Z}$, and corrupted by Gaussian noise. The attention output can thus be interpreted as the expectation of $\boldsymbol{v}$ given the query $\boldsymbol{u}$, corresponding to Bayesian denoising under the discrete mixture model. We formalise this perspective in Theorem 3.2.1 and its proof.

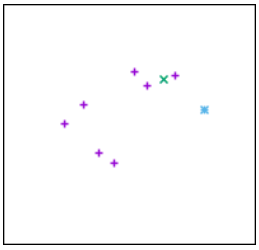


Figure 3.1: We can visualise the attention function Attn(u, $\boldsymbol{Z}$), in 2D, over the set of vectors $\boldsymbol{Z}$ with projected query u. The result is an interpolation between keys that are close to the query.

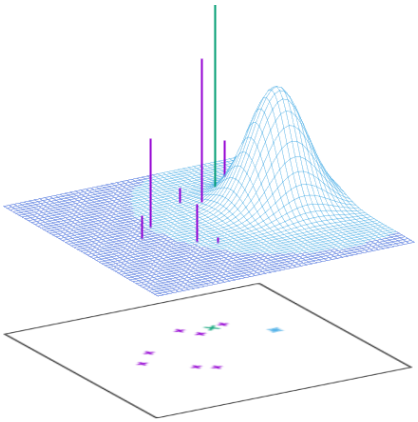


Figure 3.2: Denoising attention DAttn(u, $\boldsymbol{Z}$) over the impulse distribution $\delta_{\boldsymbol{z}_i}$ at vector $\boldsymbol{z}_i$ with query $u \sim \mathcal{N}(\boldsymbol{v}, \sqrt{d})$. The result is an expectation over the posterior given the noisy query.

Theorem 3.2.1. *Consider the attention function* $\text{Attn}(\boldsymbol{u}, \boldsymbol{Z})$ *over the set of vectors* $\boldsymbol{Z}$ *with query* $\boldsymbol{u}$*. We interpret the query as a multivariate Gaussian random variable* $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{v}, \sqrt{d}\boldsymbol{I})$ *with true mean vector* $\boldsymbol{v}$ *and diagonal covariance* $\sqrt{d}\boldsymbol{I}$*. Let* $F_{\boldsymbol{Z}}$ *denote a prior mixture distribution over* $\boldsymbol{v}$*, specified by* $\boldsymbol{Z}$*:*

$$F_{\boldsymbol{Z}} = \sum_{i=1}^{n} \pi_i \, \delta_{\boldsymbol{z}_i}, \quad \text{where } \pi_i = \frac{\exp\left(\frac{1}{2\sqrt{d}}\|\boldsymbol{z}_i\|^2\right)}{\sum_{j=1}^{n} \exp\left(\frac{1}{2\sqrt{d}}\|\boldsymbol{z}_j\|^2\right)},$$

and, where $\delta_{\boldsymbol{z}_i}$ *is the impulse distribution at each vector* $\boldsymbol{z}_i$*. The mixture weights* π_i *are defined by the scaled squared norms* $\|\boldsymbol{z}_i\|^2$*. Then, attention can be interpreted as Bayesian query denoising* $\text{DAttn}(\boldsymbol{u}; F_{\boldsymbol{Z}})$*. That is,*

$$\text{Attn}(\boldsymbol{u}, \boldsymbol{Z}) = \text{DAttn}(\boldsymbol{u};\ F_{\boldsymbol{Z}}) = \int_{\boldsymbol{v}} \frac{f_{\boldsymbol{Z}}(\boldsymbol{v})\ g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})}{\int_{\boldsymbol{v}} f_{\boldsymbol{Z}}(\boldsymbol{v})\ g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})\ d\boldsymbol{v}}\ \boldsymbol{v}\ d\boldsymbol{v}$$

where $f_{\boldsymbol{Z}}(\boldsymbol{v})$ *is the probability density function for distribution* $F_{\boldsymbol{Z}}$ *specified by* $\boldsymbol{Z}$*, and* $g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})$ *is the multivariate Gaussian likelihood of observing query* $\boldsymbol{u}$ *given mean* $\boldsymbol{v}$ *and diagonal covariance* $\sqrt{d}\boldsymbol{I}$*. The result of* $\text{DAttn}(\boldsymbol{u}, \boldsymbol{Z})$ *is the expected value of true vector* $\boldsymbol{v}$ *after seeing the noisy observation* $\boldsymbol{u}$*, which can be interpreted as a form of denoising.*

Proof.

$$
\begin{aligned}
\text{Attn}(\boldsymbol{u}, \boldsymbol{Z}) &= \sum_{i=1}^{n} \frac{\exp\left(\frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{z}_i\right)}{\sum_{j=1}^{n} \exp\left(\frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{z}_j\right)} \boldsymbol{z}_i \\
&= \sum_{i=1}^{n} \frac{\exp\left(\frac{1}{2\sqrt{d}}||\boldsymbol{z}_i||^2\right) \cdot \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{z}_i||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{z}_i\right)}{\sum_{j=1}^{n} \exp\left(\frac{1}{2\sqrt{d}}||\boldsymbol{z}_j||^2\right) \cdot \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{z}_j||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{z}_j\right)} \boldsymbol{z}_i && \text{(Add and subtract terms)} \\
&= \sum_{i=1}^{n} \frac{\exp\left(\frac{1}{2\sqrt{d}}||\boldsymbol{z}_i||^2\right) \int_{\boldsymbol{v}} \delta_{\boldsymbol{z}_i}(\boldsymbol{v}) \cdot \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{v}||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{v}\right) \boldsymbol{v}\, d\boldsymbol{v}}{\sum_{j=1}^{n} \exp\left(\frac{1}{2\sqrt{d}}||\boldsymbol{z}_j||^2\right) \int_{\boldsymbol{v}} \delta_{\boldsymbol{z}_j}(\boldsymbol{v}) \cdot \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{v}||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{v}\right) d\boldsymbol{v}} && \text{(Replace sum with Dirac integrals)} \\
&= \int_{\boldsymbol{v}} \frac{\left(\sum_{i=1}^{n} \frac{\exp\left(\frac{1}{2\sqrt{d}}||\boldsymbol{z}_i||^2\right)}{\sum_{j=1}^{n} \exp\left(\frac{1}{2\sqrt{d}}||\boldsymbol{z}_j||^2\right)} \delta_{\boldsymbol{z}_i}(\boldsymbol{v})\right) \cdot \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{v}||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{v}\right)}{\int_{\boldsymbol{v}} \left(\sum_{i=1}^{n} \frac{\exp\left(\frac{1}{2\sqrt{d}}||\boldsymbol{z}_i||^2\right)}{\sum_{j=1}^{n} \exp\left(\frac{1}{2\sqrt{d}}||\boldsymbol{z}_j||^2\right)} \delta_{\boldsymbol{z}_i}(\boldsymbol{v})\right) \cdot \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{v}||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{v}\right) d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Normalisation mixture form)} \\
&= \int_{\boldsymbol{v}} \frac{f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{v}||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{v}\right)}{\int_{\boldsymbol{v}} f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{v}||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{v}\right) d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Define } f_{\boldsymbol{Z}}\text{)} \\
&= \int_{\boldsymbol{v}} \frac{f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot \frac{1}{\sqrt{2\pi\sqrt{d}}} \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{v}||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{v} - \frac{1}{2\sqrt{d}}||\boldsymbol{u}||^2\right)}{\int_{\boldsymbol{v}} f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot \frac{1}{\sqrt{2\pi\sqrt{d}}} \exp\left(-\frac{1}{2\sqrt{d}}||\boldsymbol{v}||^2 + \frac{1}{\sqrt{d}}\boldsymbol{u}^\top \boldsymbol{v} - \frac{1}{2\sqrt{d}}||\boldsymbol{u}||^2\right) d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Multiply and divide terms)} \\
&= \int_{\boldsymbol{v}} \frac{f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot \frac{1}{\sqrt{2\pi\sqrt{d}}} \exp\left(-\frac{1}{2\sqrt{d}}(\boldsymbol{u} - \boldsymbol{v})^2\right)}{\int_{\boldsymbol{v}} f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot \frac{1}{\sqrt{2\pi\sqrt{d}}} \exp\left(-\frac{1}{2\sqrt{d}}(\boldsymbol{u} - \boldsymbol{v})^2\right) d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Complete the square)} \\
&= \int_{\boldsymbol{v}} \frac{f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})}{\int_{\boldsymbol{v}} f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})\, d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Express as Gaussian)}
\end{aligned}
$$

□

Theorem 3.2.1 shows that in the special case where $F = F_{\boldsymbol{Z}} = \sum_i \pi_i \delta_{\boldsymbol{z}_i}$ is a finite mixture of impulse distributions, $\text{DAttn}(\boldsymbol{u}; F)$ recovers $\text{Attn}(\boldsymbol{u}, \boldsymbol{Z})$, but with the bias term $\log(\pi_i)$ substituted for $\frac{1}{2\sqrt{d}}\|\boldsymbol{z}_i\|^2$. More generally, the denoising attention function

$$\text{DAttn}(\boldsymbol{u};\ F) = \int_{\boldsymbol{v}} \frac{f(\boldsymbol{v})\, g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})}{\int_{\boldsymbol{v}} f(\boldsymbol{v})\, g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})\, d\boldsymbol{v}}\, \boldsymbol{v}\, d\boldsymbol{v}$$

where $f(\cdot)$ is the probability density function of F, defines attention over any probability distribution on a vector space—not just finite sets of impulses. This formulation allows the latent space of a Transformer encoder-decoder to be treated as a mixture distribution.

Practical Formulation To make the method practical, we express denoising attention during training in terms of the query $\boldsymbol{u}$, a set of sampled key vectors $\boldsymbol{Z}$ and sampled mixture weights $\boldsymbol{\pi}$. This matches the standard attention form, with two key differences: (1) the keys $\boldsymbol{Z} \in \mathbb{R}^{(n+1)\times d}$ are sampled from a Gaussian mixture, including one sample from the prior component which is included as a key; and (2) each key receives an additive attention bias $\boldsymbol{b} \in \mathbb{R}^{(n+1)}$ based on its mixture weight. Then the denoising attention is given by:

$$\text{DAttention}(\boldsymbol{u}, \boldsymbol{Z}, \boldsymbol{\pi}) = \text{softmax}\left(\tfrac{1}{\sqrt{d}}\boldsymbol{u}\boldsymbol{Z}^\top + \boldsymbol{b}\right)\boldsymbol{Z}\boldsymbol{W}^V, \tag{3.1}$$

$$\text{where} \qquad \boldsymbol{b} = \log(\boldsymbol{\pi}) - \tfrac{1}{2\sqrt{d}}||\boldsymbol{Z}||^2.$$

The bias term $\boldsymbol{b}$ combines the log mixture weights $\log(\boldsymbol{\pi}) \in \mathbb{R}^{(n+1)}$ and a penalty proportional to the squared norms of the keys. This formulation is defined for a single query $\boldsymbol{u} \in \mathbb{R}^{1\times p}$ but extends naturally to batched queries. A full derivation and corresponding pseudocode is provided in Appendix A.5.

3.2.2 A Prior over Mixture Distributions

Given that our attention-based latent representations are formalised as mixture distributions F, a Bayesian approach requires a prior over these latent distributions. Attention-based models place no finite bound on the possible number of vectors in their set of vectors $\boldsymbol{Z}$, and thus there is no finite bound on the number of parameters needed to specify the equivalent mixture distribution F. Nonetheless, we can still specify probability distributions over this infinite space of possible distributions F, using methods from Bayesian nonparametrics. These nonparametric Bayesian methods, with exchangeable distributions, are specifically designed for modelling probability distributions over unboundedly large mixture distributions.

We base our distributions over mixture distributions on Dirichlet processes $\text{DP}(G_0, \alpha_0)$. As mentioned in Section 2.3, Dirichlet processes (DPs) are a generalisation of Dirichlet distributions to an infinite support, such as the points in a vector space. We define DP $F \sim \text{DP}(G_0, \alpha_0)$, where G_0 is the base distribution over vectors and $\alpha_0 \in \mathbb{R}$ is the concentration parameter, as the limit of a sequence of finite Dirichlet distributions. Each sample F from a DP is an infinite mixture of impulse distributions $\delta_{\boldsymbol{z}_i}$, parameterised by an infinite sequence of weight-vector pairs $\pi_i, \boldsymbol{z}_i$. This contrasts with the finite $\boldsymbol{Z}$ in attention-based representations. Having an infinite F would also cause problems in our variational Bayesian model, because VIB uses a bound on the log-likelihood (see Section 3.3.1), which Kingma and Welling (2014) showed has an error of $\text{D}_{\mathbb{KL}}(q(F|x) \,\|\, p(F|x))$ (the looseness of the bound). This would be infinite unless both the true posterior $p(F|x)$ and its approximation $q(F|x)$ generate a finite F, so we need a prior which generates finite F.

The Unbounded Dirichlet Process Prior We do not want a prior which places an apriori bound on the size of F, so we assume it is finite but unbounded, and propose a prior which is an unbounded sequence of finite approximations to a DP. We define a bounded DP $F \sim \text{BDP}(G_0, \alpha_0, \kappa_0)$ in equation 3.2 below:

$$
\begin{aligned}
F &= \sum_{i=1}^{\kappa_0} \pi_i \delta_{\boldsymbol{z}_i} \\
\boldsymbol{\pi} &\sim \mathrm{Dir}(\tfrac{\alpha_0}{\kappa_0}, \overset{\kappa_0}{\ldots}, \tfrac{\alpha_0}{\kappa_0}) \\
\boldsymbol{z}_i &\sim G_0 \quad \text{for } i = 1, \ldots, \kappa_0
\end{aligned}
\tag{3.2}
$$

Our approach to the prior is to use an unbounded but finite κ_0, so we define a distribution over approximations as κ_0 increases towards infinity. Hence, every distribution is over a finite number of vectors, but there is no finite bound on the number of vectors in all distributions. Given ϕ is some distribution over positive integers $\kappa \in \mathbb{Z}^+$, we define this unbounded DP as $\mathrm{UDP}(G_0, \alpha_0, \phi) = \mathrm{BDP}(G_0, \alpha_0, \kappa)$ where $\kappa \sim \phi$.

We use these definitions both to define a general prior over probability distributions, and to define a conditional prior for each input length. In both cases, the base distribution G_0^p is assumed to be a unit Gaussian (inspired by Kingma and Welling (2014)) and the concentration parameter α_0^p is assumed to be one.[I]

$$
\begin{aligned}
G_0^p &= \mathcal{N}(\boldsymbol{\mu}^p, \boldsymbol{I}(\boldsymbol{\sigma}^p)^2) \\
\alpha_0^p &= 1 \\
\boldsymbol{\mu}^p &= \mathbf{0} \\
\boldsymbol{\sigma}^p &= \mathbf{1}
\end{aligned}
$$

The general prior is $\mathrm{UDP}(G_0^p, \alpha_0^p, \phi^p)$, where the size distribution ϕ^p is determined empirically. The conditional prior $\mathrm{BDP}(G_0^p, \alpha_0^p, \kappa_0)$ sets the level of approximation κ_0 as a fixed function of the input length n, in particular $\kappa_0{=}(n+1)\kappa^\Delta$, where $\kappa^\Delta \in \mathbb{Z}^+$ is a hyperparameter that controls the approximation.

A Conditional Bounded DP Prior It will be useful to generalise this conditioning for the level of approximation to any conditional prior which is a fixed function of only the input length. If we know the input length n, but know nothing about the content of the text, then the distribution of vectors should stay the same as the general prior, $G_0^{p'} = G_0^p$. However, the count of observations we expect to have after an input of that length would not be α_0^p, but should include a pseudo-count $\alpha^\Delta \in \mathbb{R}_{\geq 0}$ hyperparameter for every token, and thus $\alpha_0^{p'} = \alpha_0^p + n\alpha^\Delta$. This then gives us the conditional prior given n of $\mathrm{BDP}(G_0^p, \alpha_0^{p'}, \kappa_0)$.

[I] We will use "p" and "q" superscripts to designate variables for the prior and posterior, respectively. Similarly, a zero subscript is part of the name of the variable, in contrast to positive integer subscripts which are indices.

Since a DP is a conjugate prior, we can use exact inference to compute the posterior DP from the prior DP plus a set of pseudo-observations output by the encoder. Each pseudo-observation is a real-valued pseudo-count $\alpha_i^q \in \mathbb{R}_{\geq 0}$ and a parametric distribution which represents uncertainty in the observation. We use an isotropic Gaussian, $G_i^q = \mathcal{N}(\boldsymbol{\mu}_i^q, \boldsymbol{I}(\boldsymbol{\sigma}_i^q)^2)$, as the parametric distribution, specified by a mean $\boldsymbol{\mu}_i^q \in \mathbb{R}^{1\times d}$ and a standard deviation $\boldsymbol{\sigma}_i^q \in \mathbb{R}_{>0}^{1\times d}$. Here we assume that the number of candidate pseudo-observations is the same as the length n of the input, but some of these pseudo-observations may have zero pseudo-counts and thus be effectively removed from the set.

3.2.3 A Posterior over Mixture Distributions

Due to the conjugacy of the Dirichlet Process, the posterior distribution can be computed exactly by conditioning on a set of pseudo-observations generated by the encoder (see Background Section 2.3.2). The posterior Dirichlet Process has an updated concentration parameter, given by the sum of the prior concentration α_0^p and a set of real-valued pseudo-counts $\alpha_i^q \in \mathbb{R}_{\geq 0}$. Each pseudo-observation is represented by a parametric distribution G_i^q, which captures uncertainty in the observation. We use isotropic Gaussians, $G_i^q = \mathcal{N}(\boldsymbol{\mu}_i^q, \boldsymbol{I}(\boldsymbol{\sigma}_i^q)^2)$, where $\boldsymbol{\mu}_i^q \in \mathbb{R}^{1\times d}$ is the mean and $\boldsymbol{\sigma}_i^q \in \mathbb{R}_{>0}^{1\times d}$ is the standard deviation. The posterior F is given by:

$$F \sim \mathrm{DP}(G_0^q,\ \alpha_0^q)$$

$$\alpha_0^q = \sum_{i=1}^{n+1} \alpha_i^q,$$

$$G_0^q = \sum_{i=1}^{n+1} \frac{\alpha_i^q}{\alpha_0^q} G_i^q$$

Here, n is the length of the input sequence, and we assume one pseudo-observation per input element. Some pseudo-counts α_i^q may be zero, in which case the corresponding component is effectively removed from the mixture. The $(n{+}1)^{\text{th}}$ component encodes the prior, with $\alpha_{n+1}^q = \alpha_0^p$ and $G_{n+1}^q = G_0^p$. This update mechanism smoothly interpolates between the prior and the observed data, shifting mass toward the empirical distribution while preserving prior influence.

We derive an alternative factorisation of the posterior DP (Appendix A.3) which helps with the sampling method in Section 3.3.2. We then bound this factorised DP so that it generates the same number $\kappa_0{=}(n{+}1)\kappa^{\Delta}$ of weighted vectors as the conditional prior $\mathrm{BDP}(G_0^p, \alpha_0^p, \kappa_0)$. The resulting bounded posterior $F \sim \mathrm{BFDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q, \kappa^{\Delta})$ is given in

Equation 3.3, which defines our posterior distribution $q(F|x)$.

$$
\begin{aligned}
F &= \sum_{i=1}^{n+1} \rho_i F_i \\
\boldsymbol{\rho} &\sim \mathrm{Dir}(\alpha_1^q, \ \ldots, \ \alpha_{n+1}^q) \\
F_i &\sim \mathrm{BDP}(G_i^q, \alpha_i^q, \kappa^\Delta) \quad \text{for } i = 1, \ \ldots, \ n+1
\end{aligned}
\tag{3.3}
$$

This posterior simplifies to a mixture of impulse distributions defined by:

$$
\begin{aligned}
F &= \sum_{i=1}^{n+1} \sum_{j=1}^{\kappa^\Delta} \rho_i \pi'_{ij} \delta_{\boldsymbol{z}_{ij}} \\
\boldsymbol{\rho} &\sim \mathrm{Dir}(\alpha_1^q, \ \ldots, \ \alpha_{n+1}^q) \\
\boldsymbol{\pi}'_i &\sim \mathrm{Dir}(\frac{\alpha_i^q}{\kappa^\Delta}, \ \overset{\kappa^\Delta}{\ldots}, \ \frac{\alpha_i^q}{\kappa^\Delta}) \\
\boldsymbol{z}_{ij} &\sim G_i^q.
\end{aligned}
$$

The Mean Posterior Mixture Distribution A VAE is trained on samples from the posterior, but at test time VAEs typically use the mean of this distribution. Generalising the latent space to mixture distributions makes this straightforward, since the mean of our BFDP posterior is its base distribution G_0^q. This base distribution is a continuous distribution, whereas at training time all samples are discrete distributions. Nonetheless, when accessed via denoising attention, the base distribution looks like a typical sample from the posterior.

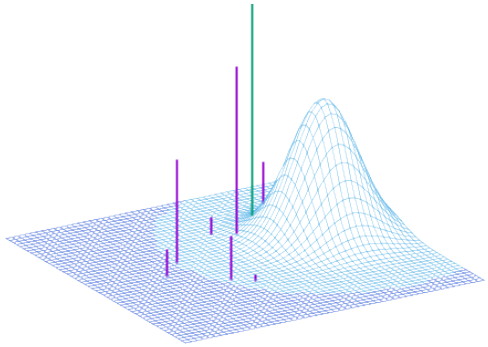


Figure 3.3: **Training-time** denoising attention $\mathrm{DAttn}(u;\ F_{\boldsymbol{Z}})$, where the discrete impulse distributions $\delta_{\boldsymbol{z}_i}$ is sampled.

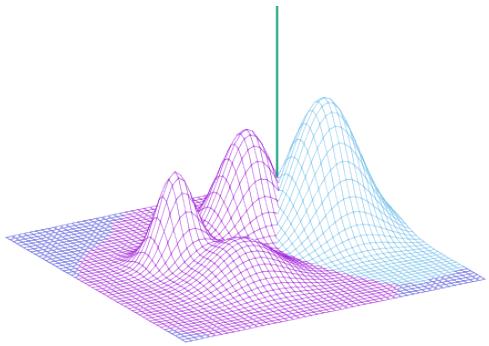


Figure 3.4: **Test-time** denoising attention $\mathrm{DAttn}(u;\ F)$ where the continuous base distribution F is a mixture of Gaussians.

We visualised this by comparing the vector returned by denoising attention (in green) given a typical sample from this distribution (Figure 3.3) and the continuous mean distribution (Figure 3.4). Thus, the function defined by applying denoising attention to a sampled distribution can be seen as a noisy version of the function defined by applying denoising attention to the mean distribution.

To compute denoising attention during test time with a Gaussian mixture base distribution G_0^q, we use the result that product of two Gaussians is another Gaussian (Petersen et al., 2008).

Theorem 3.2.2. *Let* $\text{DAttn}(\boldsymbol{u}, F_{\boldsymbol{Z}})$ *denote the denoising attention function, where* $F_{\boldsymbol{Z}}$ *is sampled during training. At evaluation time, we replace the random distribution with its expectation. For Dirichlet processes, this corresponds to using the base distribution* G_0^q*. Then, the deterministic denoising attention becomes:*

$$\text{DAttn}(\boldsymbol{u};\ G_0^q) = \sum_i \frac{\alpha_i^q\, g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))}{\sum_i \alpha_i^q\, g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))} \left(\frac{\frac{1}{\sqrt{d}}\,\boldsymbol{u} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}\,\boldsymbol{\mu}_i^q}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}} \right)$$

Proof.

$$\begin{aligned}
\text{DAttn}(\boldsymbol{u}, G_0^q) &= \int_{\boldsymbol{v}} \frac{G_0^q(\boldsymbol{v}) \cdot g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})}{\int_{\boldsymbol{v}} G_0^q(\boldsymbol{v}) \cdot g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})\, d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Base distribution)} \\
&= \int_{\boldsymbol{v}} \frac{G_0^q(\boldsymbol{v}) \cdot g(\boldsymbol{v};\ \boldsymbol{u}, \sqrt{d}\boldsymbol{I})}{\int_{\boldsymbol{v}} G_0^q(\boldsymbol{v}) \cdot g(\boldsymbol{v};\ \boldsymbol{u}, \sqrt{d}\boldsymbol{I})\, d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Gaussian symmetry)} \\
&= \int_{\boldsymbol{v}} \frac{\left(\sum_i \frac{\alpha_i^q}{\sum_i \alpha_i^q} g(\boldsymbol{v};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\boldsymbol{\sigma}_i^q)^2)\right)\, g(\boldsymbol{v};\ \boldsymbol{u}, \sqrt{d}\boldsymbol{I})}{\int_{\boldsymbol{v}} \left(\sum_i \frac{\alpha_i^q}{\sum_i \alpha_i^q} g(\boldsymbol{v};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\boldsymbol{\sigma}_i^q)^2)\right)\, g(\boldsymbol{v};\ \boldsymbol{u}, \sqrt{d}\boldsymbol{I})\, d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Expand out mixture)} \\
&= \int_{\boldsymbol{v}} \frac{\sum_i \alpha_i^q g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))\, g(\boldsymbol{v};\ (\frac{\frac{1}{\sqrt{d}}\boldsymbol{u} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}\boldsymbol{\mu}_i^q}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}}), \boldsymbol{I}(\frac{1}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}}))}{\int_{\boldsymbol{v}} \sum_i \alpha_i^q g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))\, g(\boldsymbol{v};\ (\frac{\frac{1}{\sqrt{d}}\boldsymbol{u} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}\boldsymbol{\mu}_i^q}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}}), \boldsymbol{I}(\frac{1}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}}))\, d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Product of Gaussians)}^{\text{II}} \\
&= \frac{\sum_i \alpha_i^q g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2)) \int_{\boldsymbol{v}} g(\boldsymbol{v};\ (\frac{\frac{1}{\sqrt{d}}\boldsymbol{u} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}\boldsymbol{\mu}_i^q}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}}), \boldsymbol{I}(\frac{1}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}}))}{\sum_i \alpha_i^q g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2)) \int_{\boldsymbol{v}} g(\boldsymbol{v};\ (\frac{\frac{1}{\sqrt{d}}\boldsymbol{u} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}\boldsymbol{\mu}_i^q}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}}), \boldsymbol{I}(\frac{1}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}}))\, d\boldsymbol{v}} \boldsymbol{v}\, d\boldsymbol{v} && \text{(Move integral inside)} \\
&= \sum_i \frac{\alpha_i^q g(\boldsymbol{u}\ ;\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))}{\sum_i \alpha_i^q g(\boldsymbol{u}\ ;\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))} \left(\frac{\frac{1}{\sqrt{d}}\boldsymbol{u} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}\boldsymbol{\mu}_i^q}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}} \right) && \text{(Expectation } \mathbb{E}(\boldsymbol{v})\text{)}
\end{aligned}$$

□

[II] Petersen et al. (2008, Section 8.1.8): $\mathcal{N}(x;\ \mu_a, \sigma_a^2) \cdot \mathcal{N}(x;\ \mu_b, \sigma_b^2) = d \cdot \mathcal{N}(x;\ \mu_c, \sigma_c^2)$, where

$$\mu_c = \frac{\frac{\mu_a}{\sigma_a^2} + \frac{\mu_b}{\sigma_b^2}}{\frac{1}{\sigma_a^2} + \frac{1}{\sigma_b^2}}, \quad \sigma_c^2 = \frac{1}{\left(\frac{1}{\sigma_a^2} + \frac{1}{\sigma_b^2}\right)}, \quad d = \mathcal{N}(\mu_b;\ \mu_a, \sigma_a^2 + \sigma_b^2).$$

The expression derived in Theorem 3.2.2 resembles the denoising attention mechanism, where the variance is accounted for by $\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2$ and the values are a weighted interpolation between the query $\boldsymbol{u}$ and the mixture component mean $\boldsymbol{\mu}_i^q$.

Practical Formulation To make the formulation of attention derived from Theorem 3.2.2 practical at evaluation time, we express denoising attention in terms of the query $\boldsymbol{u}$ and a set of deterministic keys derived from the learned posterior. Instead of sampling from the posterior F, we use the mean, which is the base distribution $G_0^q = \sum_i \frac{\alpha_i^q}{\alpha_0^q} \mathcal{N}(\boldsymbol{\mu}_i^q, \boldsymbol{I}(\boldsymbol{\sigma}_i^q)^2)$. The NVIB layer maps encoder outputs to mixture parameters $(\boldsymbol{\mu}^q, \boldsymbol{\sigma}^q, \frac{\boldsymbol{\alpha}^q}{\alpha_0^q})$, where $\alpha_0^q = \sum_i \alpha_i^q$ is used to get the expected mixture weights. This yields an attention function structurally similar to standard attention, but with three key differences: (1) it is computed deterministically from posterior parameters, (2) each key receives an additive bias, and (3) the value projection is an interpolation between the query and posterior mean. For convenience, we define $(\boldsymbol{\sigma}_i^r)^2 = \sqrt{d} + (\boldsymbol{\sigma}_i^q)^2$. Then test-time denoising attention is given by:

$$\text{DAttention}(\boldsymbol{u}, \boldsymbol{\mu}^q, \boldsymbol{\sigma}^r, \boldsymbol{\alpha}^q) = \text{softmax}\left(\boldsymbol{u}\left(\frac{\boldsymbol{\mu}^q}{(\boldsymbol{\sigma}^r)^2}\right)^\top + \boldsymbol{c}\right)\left(\frac{(\boldsymbol{\sigma}^q)^2}{(\boldsymbol{\sigma}^r)^2} \odot (\mathbf{1}_n^\top \boldsymbol{u}) + \frac{\sqrt{d}}{(\boldsymbol{\sigma}^r)^2} \odot \boldsymbol{\mu}^q\right)\boldsymbol{W}^V, \tag{3.4}$$

$$\text{where} \quad \boldsymbol{c} = \log\left(\frac{\boldsymbol{\alpha}^q}{\alpha_0^q}\right) - \left(\tfrac{1}{2}\left|\left|\frac{\boldsymbol{\mu}^q}{\boldsymbol{\sigma}^r}\right|\right|^2\right)^\top - \mathbf{1}_p \left(\log(\boldsymbol{\sigma}^r)\right)^\top.$$

Where $\mathbf{1}_p$ and $\mathbf{1}_n$ are a row vectors of ones of dimensionality p and n, respectively. As in training, the bias term $\boldsymbol{c}$ captures the combined influence of the log mixture weights and the scaled squared norms of the key vectors, However there is a third term which includes the influence of the variance. A caveat is that it applies only to single-head attention. Extending it to the multi-head setting is non-trivial which we address in Chapter 5. A full derivation and pseudocode is provided in Appendix A.5.

3.3 The Nonparametric Variational Information Bottleneck

By generalising attention-based representations to mixture distributions and generalising the attention function to denoising attention, we can define a VIB regulariser for attention-based architectures. Encoder-decoder models represent inputs as sets of vectors and use the attention function to map a query to a corresponding output vector. To define a VIB regulariser in this setting, we first map the set of vectors to a collection of pseudo-observations, or equivalently, to the parameters of a latent mixture distribution. We then use denoising attention to realise the query-to-output mapping, analogous to the decoder in a variational autoencoder. Then, given the nonparametric prior and posterior from Section 3.2, we can define our nonparametric VIB regulariser by specifying how to compute the KL divergence between the prior and posterior, and how to effectively

sample from the posterior for training. As far as we are aware, this proposal is the first VIB model for attention-based representations like Transformer embeddings.

The VIB layer in a VAE controls the amount of information passing through it by introducing noise according to a posterior output by the encoder, and regularises this information by minimising the KL divergence between this posterior and an uninformative prior. In our setting, the encoder outputs a set of real-valued pseudo-counts α_i^q, which act as mixture weights in the posterior Dirichlet Process. These pseudo-counts determine the relative importance of each component in the latent mixture and serve as a continuous relaxation of counts in Dirichlet updates. A known difficulty with VAEs is posterior collapse, where the posterior converges to the prior and the latent representation becomes uninformative (Bowman et al., 2016). Similarly to the *free-bits* objective proposed to address this in vector-space VAEs (Kingma et al., 2016), we regularise not towards the prior—which receives no pseudo-counts from the input—but towards a conditional prior $\mathrm{BDP}(G_0^p, \alpha_0^{p'}, \kappa_0)$, which receives $n\alpha^{\Delta}$ pseudo-counts but remains agnostic to their content. We find this improves training stability and helps avoid posterior collapse.

3.3.1 The Variational Information Bottleneck Loss

As introduced in Section 2.2.2, the Evidence Lower Bound (ELBO) approximates the log marginal likelihood of an observation x. In the Variational Information Bottleneck (VIB) framework, this bound is used to trade off predictive accuracy against compression:

$$\log p(x) \geq \underbrace{\mathbb{E}_{q(F|x)}[\log p(x|F)]}_{\mathcal{L}_R} - \mathrm{D}_{\mathbb{KL}}(q(F|x) \,||\, p(F))$$

Here, F is a latent representation of x, and $q(F|x)$ is a variational approximation to the true posterior. The first term, $\mathcal{L}_R$, is the reconstruction loss and encourages F to retain information useful for predicting x, while the second term compresses F by regularising $q(F|x)$ toward a prior $p(F)$.

The KL divergence between the prior and posterior is finite and well-defined. Specifically, they are conditioned on the same upper bound κ_0 on the number of vectors they generate. The prior is given by $p(F) = \mathrm{BDP}(G_0^p, \alpha_0^{p'}, \kappa_0)$, and the posterior by $q(F \mid x) = \mathrm{BFDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q, \kappa^{\Delta})$, where $\kappa_0 = (n+1)\kappa^{\Delta}$. The full derivation is provided in Appendix A.4. The KL divergence decomposes into two terms: $\mathcal{L}_D$, accounting for the divergence between the Dirichlet-distributed weights $\boldsymbol{\pi}$, and $\mathcal{L}_G$, capturing the divergence between the Gaussian components $\boldsymbol{Z}$.

To avoid over-regularising unused components, we approximate the KL divergence to focus only on active parts of the posterior. Using the exact KL divergence for our bounded DPs would penalise all components equally, including those with zero α_i^q, which have no effect on the posterior. Instead, we approximate a KL that regularises only components

with non-trivial weights. While marginalising over the number of such components is intractable for $\mathcal{L}_D$, the relationship is approximately linear (see Appendix A.4). We therefore substitute the expected number $\frac{\alpha_i^q}{\alpha_0^q}\kappa_0$ in place of the actual count in the KL expression. This yields the following loss terms for the KL divergence, which scale roughly linearly with κ_0. With these approximations, the KL divergence between the prior and posterior can be expressed as follows, where Γ denotes the gamma function and ψ the digamma function:

$$\mathrm{D}_{\mathbb{KL}}(q(F \mid x) \,\|\, p(F)) \approx \mathcal{L}_D + \mathcal{L}_G,$$

where:

$$\begin{aligned}\mathcal{L}_D = \log\Gamma(\alpha_0^q) - \log\Gamma(\alpha_0^{p'}) \\ + (\alpha_0^q - \alpha_0^{p'})\left(\psi\left(\tfrac{\alpha_0^q}{\kappa_0}\right) - \psi(\alpha_0^q)\right) \\ + \kappa_0\left(\log\Gamma\left(\tfrac{\alpha_0^{p'}}{\kappa_0}\right) - \log\Gamma\left(\tfrac{\alpha_0^q}{\kappa_0}\right)\right),\end{aligned} \tag{3.5}$$

$$\mathcal{L}_G = \tfrac{1}{2}\kappa_0 \sum_{i=1}^{n+1} \tfrac{\alpha_i^q}{\alpha_0^q} \sum_{h=1}^{d} \left(\tfrac{(\mu_{ih}^q - \mu_h^p)^2}{(\sigma_h^p)^2} + \tfrac{(\sigma_{ih}^q)^2}{(\sigma_h^p)^2} - 1 - \log\tfrac{(\sigma_{ih}^q)^2}{(\sigma_h^p)^2}\right). \tag{3.6}$$

The VIB perspective offers a general framework for controlling information flow through latent representations. As discussed in Section 2.2.2, this allows the ELBO to be interpreted not just as a bound on the marginal likelihood, but as a means of regularising representational capacity (Alemi et al., 2017). It also permits independent weighting of different terms in the objective. We introduce two hyperparameters to modulate the influence of the KL divergence components, resulting in the final VIB loss $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_R \;+\; \lambda_D\mathcal{L}_D \;+\; \lambda_G\mathcal{L}_G \tag{3.7}$$

3.3.2 Sampling a Mixture Distribution from the Posterior

To control the amount of information which passes from the encoder to the decoder, at training time a VAE (Kingma and Welling, 2014) samples from the encoder's posterior distribution and uses this sample to reconstruct the input. The "reparameterisation trick" is used to ensure that backpropagation of the reconstruction error through this sampling step can be done effectively. We propose a novel reparameterisation trick for bounded Dirichlet processes which allows sampling without any categorical choices, and propose specific sampling methods which result in effective backpropagation through the sampling step.

For our NVIB model, we sample the parameters $\langle \boldsymbol{\pi}, \boldsymbol{Z} \rangle$ of a mixture distribution F generated by our bounded Dirichlet process posterior $\mathrm{BFDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q, \kappa^\Delta)$, where F consists of a set of impulse distributions $\delta_{\boldsymbol{z}_i}$ each with a weight π_i. A straightforward approach to sampling from a Dirichlet process would independently sample weights $\boldsymbol{\pi}$ from a (theoretically infinite) Dirichlet distribution and sample vectors $\boldsymbol{Z}$ from the base distribution of the DP, where sampling from the base distribution involves first sampling a component of the base distribution and then sampling a vector from that component's Gaussian.

A Factorised Sampling Method The problem is that sampling a component is a discrete choice, for which there is no exact reparameterisation trick. Instead, we note that the components do not differ in the number of vectors sampled from each one (always theoretically infinite for a DP), but only differ in the distribution of weights for those vectors. As specified in the factorised DP in Section 3.2.3, we characterise this distribution over weights by factorising it into two steps: first choosing how the total weight is distributed across components ($\boldsymbol{\rho}$), and then for each component choosing how its weight is distributed across its vectors ($\boldsymbol{\pi}'_i$). These are both continuous choices. The vectors can then be sampled independently from each component.

3.3.3 Reparameterisation Tricks

To enable gradient-based optimisation through sampling, we use reparameterisation tricks, as introduced in Section 2.2.2. Our model samples both Gaussian latent vectors and Dirichlet-distributed weights. These require different methods, which we describe below. Figure 3.5 summarises the strategies used.

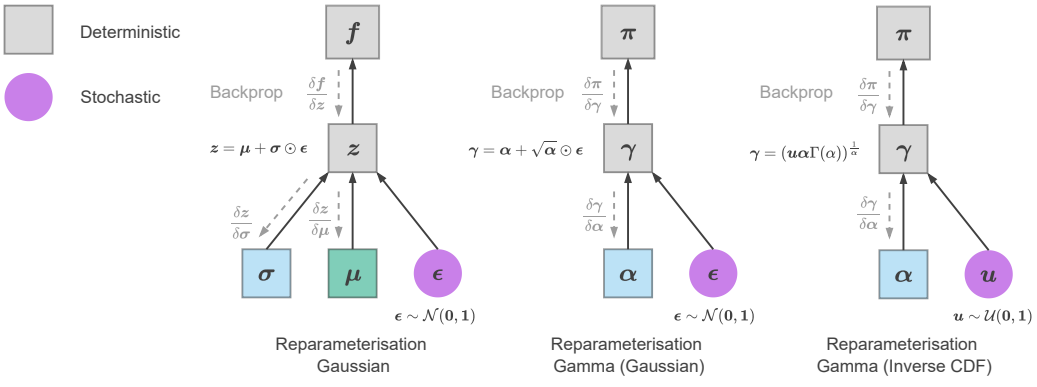


Figure 3.5: Reparameterisation tricks used for Gaussian latent variables $\boldsymbol{z}$ and Dirichlet weights $\boldsymbol{\pi}$ sampled and normalised from Gamma variables $\boldsymbol{\gamma}$.

Gaussian Components of the Base Distribution Each individual component G_i^q of the base distribution G_0^q is modelled as a diagonal-covariance Gaussian. To sample a vector $\boldsymbol{z}_k$ from component i, we use the standard location-scale reparameterisation trick (Kingma and Welling, 2014), which allows gradients flow through $\boldsymbol{\mu}_i^q$ and $\boldsymbol{\sigma}_i^q$ while treating ϵ_k as independent noise.

$$\boldsymbol{z}_k = \boldsymbol{\mu}_i^q + \boldsymbol{\sigma}_i^q \, \epsilon_k,$$

$$\epsilon_k \sim \mathcal{N}(\mathbf{0}, \mathbf{1}).$$

Dirichlet-Distributed Mixture Weights We model the mixture weights $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_\kappa)$ as a sample from a Dirichlet distribution with parameter vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_\kappa)$. This can be implemented by drawing auxiliary variables $\gamma_i \sim \Gamma(\alpha_i, 1)$ and normalising:

$$\pi_i = \frac{\gamma_i}{\sum_{j=1}^{\kappa} \gamma_j},$$

$$\gamma_i \sim \Gamma(\alpha_i, \beta = 1).$$

Since there is no exact, closed-form reparameterisation for $\Gamma(\alpha_i, 1)$, we follow Knowles (2015) and adopt two complementary approximations:

$$\gamma_i \approx \left(u_i \, \alpha_i \, \Gamma(\alpha_i)\right)^{1/\alpha_i}, \qquad (3.8)$$

$$u_i \sim \mathrm{U}(0, 1)$$

$$\gamma_i \approx \alpha_i + \sqrt{\alpha_i}\, \epsilon_i, \qquad (3.9)$$

$$\epsilon_i \sim \mathcal{N}(0, 1)$$

In Equation (3.8), the inverse-CDF approximation is accurate when $\alpha_i < 1$, but its error grows as α_i increases. Conversely, the Gaussian approximation in Equation (3.9) becomes reliable for large α_i, yet it may yield negative samples if α_i is small.

As shown in Figure 3.6, we switch at $\alpha_i = 0.6363$, truncating negative Gaussian draws to zero. This yields a piecewise reparameterisation practical across all α_i.

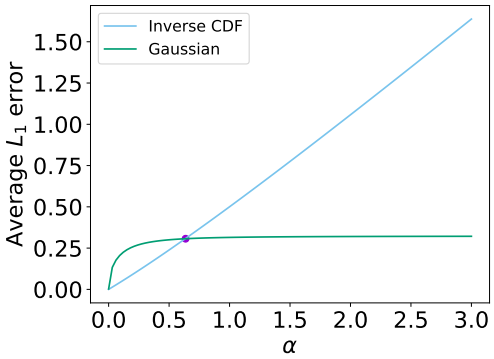


Figure 3.6: Average L_1 error between the true Gamma inverse CDF and its approximations. The crossover is at $\alpha \approx 0.6363$.

3.4 The Nonparametric Variational Autoencoder

We define a VAE for Transformers by applying the nonparametric VIB from Section 3.3. This regularises the attention-based representation between the Transformer encoder and decoder, as shown in Figure 3.7. The Transformer encoder estimates the posterior parameters $\langle \boldsymbol{\alpha}^q, \boldsymbol{\mu}^q, \boldsymbol{\sigma}^q \rangle$ given the input text x. The Transformer decoder reconstructs x using denoising cross attention over a sample F drawn from this posterior.

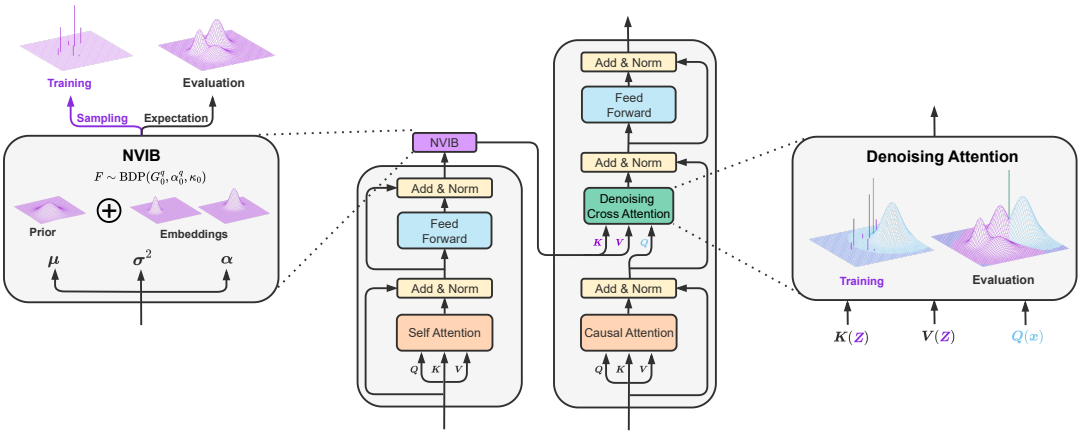


Figure 3.7: Illustration of the NVAE Transformer encoder-decoder architecture. **Center:** Main encoder-decoder structure. **Left:** NVIB layer, which maps encoder outputs to the parameters of a Bounded Dirichlet Process and appends the prior component embedding. During training, samples are drawn; at evaluation, the expected base distribution is used. **Right:** Denoising Attention module, which performs Bayesian Query Denoising using the noisy query and the mixture distribution F. Sampled vectors serve as keys and values, and mixture weights are integrated as a bias term in the attention function.

The NVIB Regulariser The NVIB layer regularises the flow of information from the Transformer encoder to the decoder, as shown in Figure 3.7. As in the fixed vector-space VIB, the KL divergence term $\mathcal{L}_G$ (Equation 3.6) encourages smaller $\boldsymbol{\mu}_i^q$ and larger $\boldsymbol{\sigma}_i^q$, effectively increasing the noise of individual latent components. In the nonparametric case, the KL term $\mathcal{L}_D$ (Equation 3.5) also promotes smaller and sparser mixture weights α_i^q, reducing the total mass of the posterior and increasing uncertainty in the mixture. This shifts probability mass toward the prior and decreases the number of active components. The optimisation concentrates information into a small subset of components with higher certainty, effectively inducing sparsity. This mechanism allows the KL regulariser to control both the content (via $\boldsymbol{\sigma}_i^q$) and the number of active vectors (via α_i^q) in the latent representation.

The Transformer Encoder The encoder maps the input text to a posterior over latent components, as illustrated in Figure 3.7. Given an input x with n tokens, it defines the posterior $q(F \mid x)$ by computing an embedding for each token i. Each embedding is projected to three parameters: $\alpha_i^q \in \mathbb{R}$, $\boldsymbol{\mu}_i^q \in \mathbb{R}^{1\times p}$, and $\log(\boldsymbol{\sigma}_i^q) \in \mathbb{R}^{1\times p}$. The variance parameters are exponentiated to ensure positivity, while the pseudo-counts α_i^q are computed using a ReLU activation (Nair and Hinton, 2010), which masks the vector during cross-attention when zero. The resulting Dirichlet process posterior includes one component $\langle \alpha_i^q, \boldsymbol{\mu}_i^q, \boldsymbol{\sigma}_i^q \rangle$ per token, plus an additional prior component, concatenated to form $n+1$ total components. During training, the mixture is sampled $\kappa_0 = (n+1)\kappa^\Delta$ times, where $\kappa^\Delta \in \mathbb{Z}^+$ is a hyperparameter controlling the number of samples per component.

The Transformer Decoder The Transformer decoder defines the likelihood $q(x \mid F)$ by conditioning on the mixture distribution F over vectors and reconstructing the input text x. During training, F is specified by sampled vectors $\mathbf{Z} \in \mathbb{R}^{\kappa_0\times p}$ and weights $\boldsymbol{\pi} \in \mathbb{R}^{\kappa_0\times 1}$. At test time, F is specified by the encoder outputs: $\boldsymbol{\alpha}^q \in \mathbb{R}^{\kappa_0\times 1}$, $\boldsymbol{\mu}^q \in \mathbb{R}^{\kappa_0\times p}$, and $\log(\boldsymbol{\sigma}^q) \in \mathbb{R}^{\kappa_0\times p}$. In both cases, the decoder accesses F using denoising cross attention, analogous to standard cross-attention in Transformer decoders. Text generation uses teacher forcing during training. At test time, decoding proceeds autoregressively using greedy decoding, terminating either upon generating an end-of-sequence token or when the output exceeds the target length by 50 tokens.

The Generative Model Our NVAE model can be used as a generative model of text. We sample from the prior and decode using the trained Transformer decoder. As discussed in Section 3.2.2, we first sample a sentence length, then draw from the conditional bounded DP prior given that length. To simplify this step, we first sample lengths from the empirical distribution of sentence lengths in the training data.

3.5 Evaluation of NVIB in NVAE

To support our theoretical contributions, we provide proof-of-concept experiments which demonstrate that our proposed NVIB regulariser performs as claimed. We evaluate it in our proposed NVAE model by training NVAEs on natural language text and evaluating the resulting models. We show that the NVAE is a viable VAE model as it exhibits a competitive reconstruction versus generation trade-off (Section 3.5.1). We show that the NVIB layer is able to dynamically regularise and choose the number of components it needs in its embeddings (Section 3.5.2). Additionally, NVIB provides an intuitive way to interpolate between sentence embeddings, which provides an evaluation of the smoothness of the latent space (Section 3.5.3).

Data We use the Wikitext-2 and Wikitext-103 datasets (Merity et al., 2017), which are high-quality English Wikipedia corpora of small and large scale, respectively. Wikitext-2, a subset of Wikitext-103, is used for most experiments, while the larger dataset is reserved for the larger-model experiments and interpolation analysis (Section 3.5.3). All text is cleaned and segmented at the sentence level using the NLTK toolkit (Bird et al., 2009), and we retain only inputs with 5 to 50 wordpiece tokens based on the BERT tokenizer (Devlin et al., 2019). Dataset statistics are shown in Table 3.1.

Table 3.1: Dataset statistics. Number of sentence examples in the train, validation and test sets. Number of word piece tokens per sentence example.

	Train/Val/Test	**Tokens**
Wikitext-2	$77K/8K/9K$	26 ± 12
Wikitext-103	$3578K/9K/8K$	25 ± 10

Baselines Models We compare to various alternative ways to define a VAE from a Transformer autoencoder. As representative of a standard fixed-length-vector VAE, the Variational Transformer Pooled (VTP) baseline pools its vectors across the sequence length dimension, and then applies a Gaussian VIB layer (Kingma and Welling, 2014). At the other extreme, the Variational Transformer (VT) baseline keeps all its vectors and applies a Gaussian VIB layer to each one. In between these baselines, as a hand-coded solution to constraining the quantity of latent vectors, Variational Transformer Stride (VTS) baselines, with parameter S, masks $1-S$ proportion of the embedding vectors based on their position. For comparability, all our baselines only differ from the NVAE model in the latent representation between the encoder and decoder, with the same Transformer encoder and Transformer decoder architectures.

Training Details We train all models from scratch using a consistent architecture for fair comparison. We use a two layer Transformer encoder and decoder with a single attention-head. The size for the word embedding vectors and model projections are 256, feed forward dimensions 1024, which leads to models of approximately 19 million trainable parameters. The BERT base-uncased tokeniser is used for tokenisation with a vocabulary of approximately 30K. During training we use: a constant learning rate of $1e^{-4}$, Adam optimiser (Kingma and Ba, 2015), a batch size of 256, gradient norm clipping 0.1 and trained for 50 epochs ($\approx 15K$ steps). The number of epochs were selected considering model convergence and minimising computation time. As a form of regularisation we use a dropout rate of 0.1 and the VIB hyperparameters λ_G, λ_D, α^{Δ} and κ^{Δ} are selected through hyperparameter tuning. We experimented with learning rate schedules, KL annealing, and free-bits objectives, but found them unnecessary for convergence. Each model experiment takes approximately 2hrs to run on a single NVIDIA GeForce RTX 3090.

VIB Hyperparameters We normalise KL weights to remove undesirable dependencies on sentence length and dimensionality. Since $\mathcal{L}_D$ and $\mathcal{L}_G$ scale roughly linearly with sentence length n, we set their weights inversely proportional to n. Additionally, we scale the Gaussian KL weight by $\frac{1}{d}$, where d is the dimensionality, to eliminate scaling with vector size. Specifically, we set $\lambda_D = \frac{\lambda'_D}{n}$ and $\lambda_G = \frac{\lambda'_G}{nd}$, where λ'_D and λ'_G are fixed hyperparameters. We also consider α^{Δ}, the conditional prior on pseudo-counts, and κ^{Δ}, the number of samples per component.

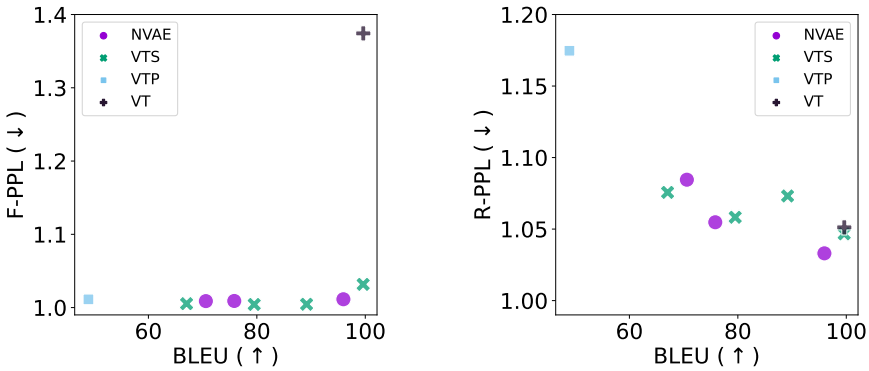
Generation Metrics We use forward and reverse perplexity to evaluate generation quality (Zhao et al., 2018; Cífka et al., 2018) . Both metrics rely on an external language model, which we implement as a Transformer with the same configuration as our main models, but without VIB regularisation. We generate 100K sentences from the model under evaluation. Forward perplexity (F-PPL) measures fluency: it is the perplexity of the external model trained on the training data and evaluated on generated text. Reverse perplexity (R-PPL) measures distributional similarity: it is the perplexity of the external model trained on generated samples and evaluated on the validation or test set.

3.5.1 Reconstruction versus Generation

All models, including baselines, are tuned on the validation set (Appendix A.1) to select optimal hyperparameters across 5 random seeds. Final results are reported on the WikiText-2 test set. For baselines, we vary the VTS parameter S, which controls the masking ratio, and the VTP pooling type. Reconstruction is evaluated using SacreBLEU (Papineni et al., 2002; Post, 2018), which measures n-gram overlap between the input and reconstructed output. Generation quality is assessed using forward perplexity (F-PPL) and reverse perplexity (R-PPL) (Zhao et al., 2018; Cífka et al., 2018).

The trade-off between reconstruction and generation is shown in Figures 3.8a and 3.8b, where lower right is better. Full validation and test results are available in Appendix Tables A.1 and A.2. The single-vector baseline (VTP) fails to reconstruct accurately (low BLEU) or generate diverse sentences (high R-PPL). The full-vector baseline (VT) struggles with fluency (high F-PPL), while the position-based masking baseline (VTS) demonstrates that regularising the latent space improves both reconstruction and generation.

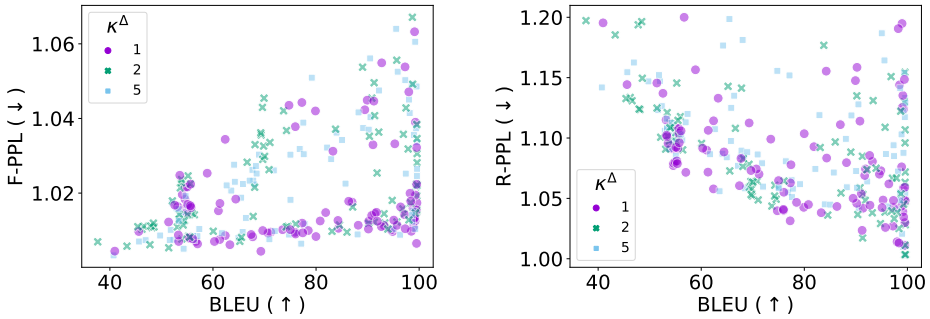
The NVAEs strike a strong balance between reconstruction and generation. The best NVAE models match or outperform the strongest VTS baselines, particularly in generation, while maintaining high reconstruction accuracy. Low R-PPL scores confirm that NVAEs sample diverse sentences, and low F-PPL indicates these are fluent (see samples in Appendix A.6). We attribute this to learning which latent vectors to retain, as opposed to relying on hand-crafted position-based dropout.



(a) F-PPL vs test BLEU. (b) R-PPL vs test BLEU.

Figure 3.8: Reconstruction versus generation trade-off.

Sampling Approximation Analysis We investigate the effect of the number of samples $\kappa^{\Delta} \in \{1, 2, 5\}$ per component, using a subset of hyperparameters: $\lambda'_G \in \{10^{-3}, 10^{-4}\}$, $\lambda'_D \in \{1, 10\}$, and $\alpha^{\Delta} \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.75, 1\}$, each evaluated across 5 random seeds. Figure 3.9 shows validation F-PPL and R-PPL versus reconstruction BLEU for different values of κ^{Δ}. We observe no significant improvement in generation or reconstruction quality with more samples. For efficiency, we therefore use a single sample per component ($\kappa^{\Delta} = 1$) in all other experiments.



(a) Validation BLEU vs F-PPL. (b) Validation BLEU vs R-PPL.

Figure 3.9: Number of samples κ^{Δ} effect on reconstruction and generation trade off.

Scaling to Larger Models and Datasets We conduct a limited larger scale experiment on Wikitext-103, using six-layer Transformer encoders and decoders with 512-dimensional embeddings and 2048 feed-forward layers (≈76M parameters). This is inline with the Transformer base size (Vaswani et al., 2017). Training is performed for 11 epochs (≈ 150K steps) with a constant learning rate of 1×10^{-5}, using a single NVIDIA Tesla V100 GPU (≈24h per run). We first train NVAE using the best-performing hyperparameters from Table A.2 (lowest R-PPL), and then select a VTS baseline with similar vector usage ν

for fair comparison. Table 3.2 shows that while VTP improves reconstruction with scale, it still performs poorly in generation. In contrast, NVAE dynamically adjusts vector usage and remains competitive with all baselines across reconstruction and generation metrics.

Table 3.2: Larger scale model results for regularisation and generation on the Wikitext-103 test set.

			Reconstruction		Generation	
Model		ν	**BLEU** (↑)	**PPL** (↓)	**F-PPL** (↓)	**R-PPL** (↓)
VT		1.00	99.56	1.00	1.06	1.08
VTP	$P = max$	0.05	97.42	1.54	2.41	1.01
VTS	$S = 0.8$	0.20	99.52	1.00	1.04	1.01
NVAE	$\alpha^{\Delta} = 0.4$	0.15	99.51	1.01	1.01	1.01

3.5.2 Regularisation

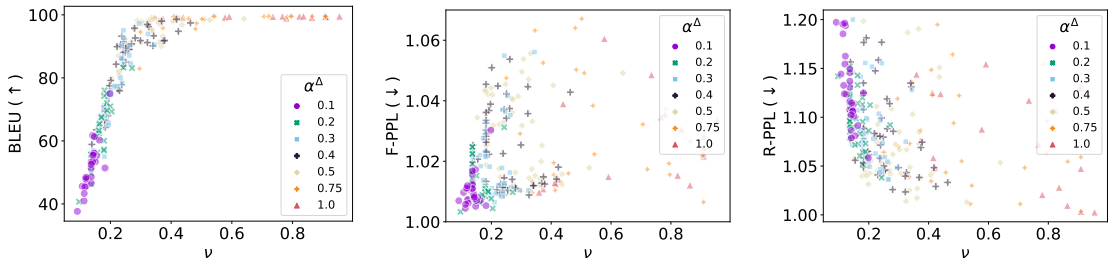
The NVIB layer enables dynamic and controllable regularisation of the latent vector set. Without regularisation, the NVAE reduces to a standard Transformer with no noise and full vector retention, as in standard VAEs. The conditional prior hyperparameter α^{Δ} provides fine-grained control over sparsity, allowing the model to balance compression and expressivity. Finally, the NVIB layer learns dynamic sparsity patterns that adapt to sequence length, selectively retaining latent vectors within the NVAE.

Without regularisation We examine the NVAE's behaviour without regularisation. Table 3.3 shows the model ignores noise and reverts to a standard Transformer, retaining all vectors ($\nu = 1$) and achieving near-perfect reconstruction. This behaviour is expected and typical of a VAE without regularisation.

Table 3.3: Reconstruction results on Wikitext-2 validation data without regularisation.

		Reconstruction	
Model	ν	**BLEU** (↑)	**PPL** (↓)
T	1	99.63 ±0.00	1.00 ±0.00
VT	1	99.63 ±0.00	1.00 ±0.00
NVAE	1	99.63 ±0.00	1.00 ±0.00

Conditional Prior Analysis We analyse the impact of the conditional prior hyperparameter α^{Δ} on vector retention. Using a subset of hyperparameters with strong reconstruction and generation performance—$\lambda'_G = \{1e^{-3}, 1e^{-4}\}$, $\lambda'_D = \{1, 10\}$, and $\kappa^{\Delta} = \{1, 2, 5\}$ across 5 random seeds—Figure 3.10 shows validation reconstruction and generation metrics plotted against the average proportion ν of vectors retained. Results indicate that α^{Δ} closely controls ν, demonstrating that the conditional prior effectively regulates sparsity in the latent representation.



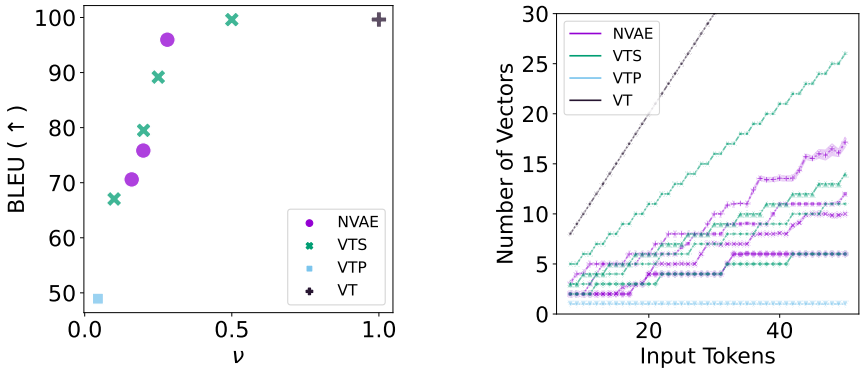
(a) Validation BLEU vs proportion of retained vectors ν.

(b) Validation F-PPL vs proportion of retained vectors ν.

(c) Validation R-PPL vs proportion of retained vectors ν.

Figure 3.10: Conditional prior α^Δ controlling the proportion of retained vectors ν.

Dynamic Regularisation The conditional prior hyperparameter α^Δ controls the proportion of latent vectors retained by the NVAE. As shown in Figure 3.11a, certain values of α^Δ enable the model to discard a significant fraction of vectors $(1-\nu)$ while preserving high reconstruction quality. Moreover, Figure 3.11b demonstrates that the NVAE adapts the number of retained vectors dynamically based on input length and content, without relying on heuristics like VTS. Larger α^Δ values correspond to higher vector retention, confirming that this hyperparameter offers fine-grained control over sparsity in the latent space.



(a) Test BLEU vs proportion of retained vectors ν.

(b) Latent quantity of vectors vs input tokens.

Figure 3.11: Regularisation analysis of the number of latent vectors.

3.5.3 Interpolation

Interpolation evaluates the smoothness of the latent space, reflecting how well the model captures the underlying data and structures the latent space. However, interpolation becomes challenging when dealing with sets of latent vectors, especially when their sizes differ. The NVIB framework addresses this by using mixture distributions, offering a more flexible, abstract latent space representation. It avoids the issue of aligning vectors

(Discussed further in Appendix A.2) and handles varying set sizes by interpolating the mixture weights across the latent space.

Given two latent mixture distributions, F_1 and F_2, we decode from the combined mixture distribution $(\tau F_1 + (1-\tau)F_2)$ for varying interpolation rates $0 \leq \tau \leq 1$. In contrast, the baselines use $\tau \mathbf{Z}_1 + (1-\tau)\mathbf{Z}_2$, where interpolation occurs over the content of the latent vectors $\mathbf{Z}_i$. For these baselines, latent sets are aligned by input position, with smaller sets padded by zero vectors, the mean of the Gaussian prior.

Table 3.4: The proportion of interpolations different from $\mathbf{S}_1$ and $\mathbf{S}_2$ by varying the interpolation rate τ. Fluency metric F-PPL of interpolations when $\tau = 0.5$.

		τ			
Model		**0.25**	**0.5**	**0.75**	**F-PPL** (↓)
VT		0.04	1.00	0.04	$1 + 1.6e^{-3}$
VTP	$P = max$	0.57	0.99	0.57	$1 + 3.3e^{-4}$
VTS	$S = 0.8$	0.04	1.00	0.04	$1 + 2.8e^{-3}$
NVAE	$\alpha^{\Delta} = 0.4$	**0.88**	0.89	**0.88**	$\mathbf{1 + 1.3e^{-7}}$

To evaluate interpolation quality, we use large-scale models, as shown in Table 3.2, and the Wikitext-103 validation set for sentences $\mathbf{S}_1$, with their reverse order for $\mathbf{S}_2$. We then interpolate between these sentences. The results, presented in Table 3.4 and Figure 3.12, show that the NVIB regulariser in NVAE yields smoother interpolations and improves fluency compared to the baselines. F-PPL is calculated across interpolations using a Transformer language model trained on Wikitext-103 at the same scale as the larger models. To avoid biasing the F-PPL metric, we exclude collapses to exactly $\mathbf{S}_1$ or $\mathbf{S}_2$. These findings are supported by qualitative examples in Appendix A.7.

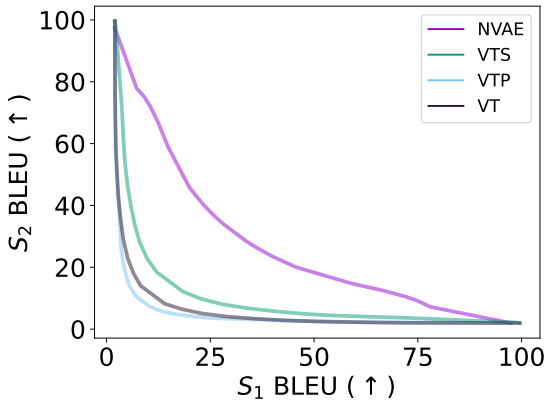


Figure 3.12: BLEU with $\mathbf{S}_1$ versus $\mathbf{S}_2$ for varying interpolations τ.

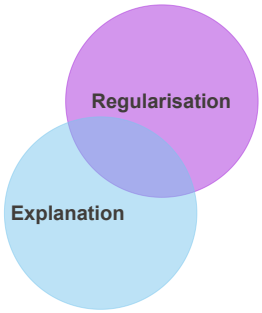
3.6 Conclusion

This chapter generalises the latent representations of attention-based models to mixture distributions over a vector space. It introduces a nonparametric variational information bottleneck to regularise these representations. We propose the nonparametric VAE, using a Transformer encoder to embed text in a nonparametric space of distributions over mixture distributions, and a Transformer decoder to generate text from a sampled mixture distribution. This Bayesian formalisation captures two key properties of the attention function: its invariance to permutations of input vectors and its ability to handle inputs of varying sizes. The NVIB model further regularises attention-based representations, ensuring the representation size aligns with the input's complexity. This is crucial for encoding text, where input sizes can vary widely.

Empirical evaluations demonstrate that NVAE performs competitively as a VAE. It successfully reconstructs input sentences and generates a coherent distribution over sentences from the prior. The model effectively regularises the size of latent representations and ensures a smooth latent space by interpolating intuitively between latent mixture distributions.

In the next chapter, we build upon these concepts by exploring how sparsity in the latent space relates to abstraction. Intuitively, using fewer latent vectors encourages each to represent more general, abstract features. We investigate how such sparse abstractions can be induced in the stacked self-attention layers of Transformers. This addresses the open question of how abstraction emerges in Transformer models. We examine whether these compressed representations perform well on downstream tasks and whether they are linguistically meaningful or interpretable.

4 Abstraction in Transformers with NVIB



Research Question

How can hierarchical and interpretable abstraction of information be induced in Transformers?

Chapter Summary

Transformers excel at modelling language but lack mechanisms to induce interpretable abstractions. Our method applies Nonparametric Variational Information Bottleneck (NVIB) across stacked self-attention layers, encouraging progressively sparser and more compressed representations at greater depth. This structure allows the model to learn a natural hierarchy of linguistic abstraction, from characters to subwords to words, without relying on tokenisation or external supervision. The resulting representations are more interpretable, capture stronger linguistic features, and exhibit improved robustness to adversarial perturbations.

Publication

- "Learning to Abstract with Nonparametric Variational Information Bottleneck"
  Behjati M.*, **Fehr F.***, Henderson J.
  EMNLP 2023 Findings

Code Repository

The code is publically available at:

- https://github.com/idiap/nvib
- https://github.com/idiap/nvib_selfattention.

Author Contributions

This paper was an equal contribution between Melika Behjati and Fabio Fehr. Melika Behjati is responsible for evaluations on linguistic probing and robustness. Fabio Fehr was responsible for the model design, training and visualisations.

4.1 Learning Abstract Representations

Learning representations of language using self-supervision has become a cornerstone of NLP (Pennington et al., 2014; Peters et al., 2018; Devlin et al., 2019, *inter alia*). However, these representations are specific to their tokenisation (e.g. Byte-Pair Sennrich et al. (2016), WordPiece Schuster and Nakajima (2012), SentencePiece Kudo and Richardson (2018), characters Al-Rfou et al. (2019), and even bytes Xue et al. (2022)), which restricts the level of abstraction from the input text which their representations are able to convey. Work like CANINE Clark et al. (2022) and Charformer Tay et al. (2022) avoid problems with tokenisation by modelling individual characters or bytes, and thereafter use a stride-based downsampling to reduce the representation length. The stride pattern is fixed and thus can't be considered as learning to abstract. Behjati and Henderson (2023) recently introduced the task of learning a higher level of abstraction in a set-of-vector space by proposing Dynamic Capacity Slot Attention. In this chapter, we propose a novel character-level model of representation learning which learns different levels of abstraction in different layers of the same model.

Contributions We adapt the Nonparametric Variational Information Bottleneck (NVIB) regulariser, introduced in Chapter 3 (Henderson and Fehr, 2023), for use in the self-attention layers of a Transformer encoder. This induces abstraction by selectively dropping vectors at deeper layers, resulting in a hierarchy of representations. The learned units are intuitive and often align with words. Through a range of analyses, we show that the model captures more semantically and linguistically meaningful information than a standard Transformer baseline. It also demonstrates greater robustness to adversarial perturbations.

Related Work Modelling language at the level of characters has the advantage of providing an end-to-end framework for the models to operate, without the need for tokenization as a preprocessing step Xue et al. (2022); Ataman et al. (2020); Choe et al. (2019); Al-Rfou et al. (2019); Kawakami et al. (2017). This is at the cost of longer sequence lengths and the need for greater model depth to reach the understanding level of subword-based models. While CANINE Clark et al. (2022) and Charformer Tay et al. (2022) are some attempts to bypass these shortcomings, they do so by fixed architectural design choices. Our work differs in that it allows the model to learn how to abstract and compress the input without a hard-coded abstraction structure. Our inspiration comes from Behjati and Henderson (2023) who introduced the task of learning a higher level of abstraction and proposed a method based on Slot Attention Locatello et al. (2020) for this purpose. Our work is also related to HM-RNNs Chung et al. (2017) as it tends to learn a hierarchy of units within its layers, though it does not make discrete decisions on unit boundaries. Our approach to learning meaningful disentangled abstractions by encouraging the models to learn compressed representations through a bottleneck is shared with VAEs Kingma and Welling (2014) and other work in that line Alemi et al. (2017); Higgins et al. (2017).

4.2 The Hierarchical NVIB Encoder

We aim to induce hierarchical abstraction in Transformers. To do this, we pretrain a Transformer autoencoder using a denoising objective, following Lewis et al. (2020). The model consists of standard Transformer encoder-decoder blocks (Vaswani et al., 2017), with the encoder modified to include NVIB regularisation in its self-attention layers. This encoder architecture is shown in Figure 4.1.

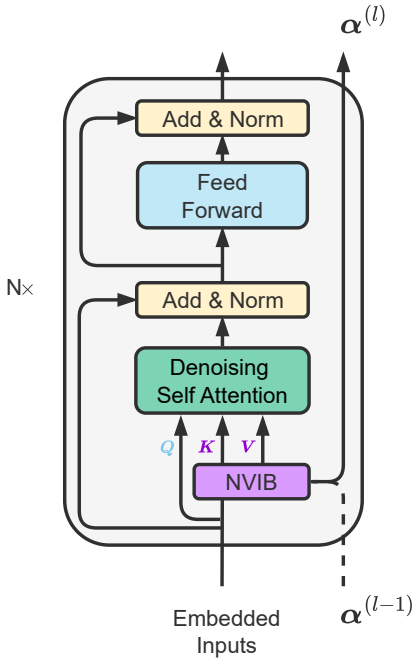


Figure 4.1: Transformer encoder layer (l) with NVIB regularisation applied to self-attention.

4.2.1 Self-Attention with NVIB

We extend NVIB to self-attention for latent abstraction in Transformer encoders. The Nonparametric Variational Information Bottleneck (NVIB), introduced in Chapter 3 (Henderson and Fehr, 2023), is an information-theoretic regulariser for attention-based latent representations. It was previously applied to cross-attention in encoder-decoder models, where it produced smooth and sparse representations. It does this by generalising attention over a set of vectors to *denoising attention* over a mixture of impulse distributions, where added noise removes information. We adapt this to stacked self-attention layers and incorporate implicit reparameterisation gradients (Figurnov et al., 2018), exponential activations, and a multiplicative skip connection. This enables the model to learn increasingly abstract and interpretable latent units across layers.

The NVIB Layer To be self-contained, we recap the NVIB layer introduced in Chapter 3. It transforms an input set of n vectors $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$ into the parameters of a Dirichlet Process, defined by a total pseudo-count and a Gaussian mixture as its base distribution. Each vector is projected to a pseudo-count $\boldsymbol{\alpha} \in \mathbb{R}^n$ and a Gaussian component ($\boldsymbol{\mu} \in \mathbb{R}^{n \times p}, \boldsymbol{\sigma} \in \mathbb{R}^{n \times p}$). The model can drop vectors by setting their pseudo-counts to zero, inducing sparsity. An additional $(n{+}1)^{\text{th}}$ component encodes the prior, with fixed parameters $\alpha^p = 1$, $\boldsymbol{\mu}^p = \mathbf{0}$, and $\boldsymbol{\sigma}^p = \mathbf{1}$. The pseudo-counts are summed to compute the DP's total pseudo-count and then normalised to weight the components of the DP's base distribution. At training time, denoising attention samples from the DP. At test time, it uses the mean of the base distribution directly. In Chapter 3 we applied this to cross-attention in encoder-decoder models, where the keys and values come from the encoder and the query comes from the decoder.

Adapting NVIB for Self-Attention A key difference in self-attention is that the keys, values, and queries all come from the same set of encoder embeddings. To compute self-attention, the Dirichlet Process is formed from the parameters projected by all input vectors. A single sample from this DP is then shared across all vectors via denoising attention. Queries are computed from the original inputs $\boldsymbol{Z} \in \mathbb{R}^{n \times p}$, before the NVIB layer. We introduce key architectural and training improvements to the NVIB layer. While in Chapter 3 Henderson and Fehr (2023) use ReLU, linear, and exponential activations to compute $\boldsymbol{\alpha}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}$ respectively, we replace the activation for $\boldsymbol{\alpha}$ with an exponential function. We also introduce a multiplicative skip connection from the previous layer $l{-}1$, as shown in Figure 4.1:

$$\boldsymbol{\alpha}^{(l)} = \exp(\boldsymbol{w}\boldsymbol{Z}^\top + b + \log(\boldsymbol{\alpha}^{(l-1)})), \tag{4.1}$$

where $\boldsymbol{w} \in \mathbb{R}^{1 \times p}$ and $b \in \mathbb{R}$ define the linear projection. The exponential activation improves training stability. We threshold small values to induce sparsity, since the exponential is never exactly zero. The skip connection helps coordinate vector importance across layers and supports the emergence of hierarchy. Keeping $\boldsymbol{\alpha}$ in log-space avoids overflow and improves numerical precision. Together, these changes yield a stable multiplicative skip connection that enhances inter-layer communication. We also apply implicit reparameterisation gradients Figurnov et al. (2018), an unbiased and tractable method for backpropagation through the sampling step that avoids costly inversion of the CDF and complex approximations, as previously discussed in Section 3.3.3.

4.2.2 Layer-wise Loss for Abstraction

The NVIB loss encourages sparse, highly informative representations. As described in Chapter 3, the objective (Equation 3.7) combines three terms: a reconstruction loss $\mathcal{L}_R$, and two KL divergences, $\mathcal{L}_D$ (Equation 3.5) and $\mathcal{L}_G$ (Equation 3.6). The reconstruction

loss $\mathcal{L}_R$ is the supervised objective. It ensures that the latent representation retains enough information to predict the original text. The KL term $\mathcal{L}_G$ maintains noise in the Gaussian components, creating a bottleneck that forces the model to preserve only the most informative features, as in vector-space VAEs (Kingma and Welling, 2014). The KL term $\mathcal{L}_D$ penalises the total pseudo-count. This pushes some pseudo-counts to zero, effectively dropping their vectors and reducing the number of active components in the latent space.

We encourage abstraction by increasing regularisation at higher layers. To apply NVIB across stacked self-attention, lower layers are allowed more vectors, while upper layers are pushed to compress. We achieve this by weighting the loss terms per layer:

$$\mathcal{L} = \mathcal{L}_R \ + \ \beta^{(l)}(\lambda_D \mathcal{L}_D \ + \ \lambda_G \mathcal{L}_G) \tag{4.2}$$

$$\beta^{(l)} = \frac{l}{\sum_{l=0}^{N} l} \quad \text{for } l \in \{1, \ ..., \ N\} \tag{4.3}$$

Here, $\beta^{(l)}$ increases linearly with layer index l relative to the total number of layers N, scaling the strength of NVIB regularisation. If a vector is dropped in the final self-attention layer (i.e. zero pseudo-count), it is also dropped from the decoder's cross-attention. No other NVIB regularisation is applied to the cross-attention.

We tested alternative schedules for $\beta^{(l)}$, including uniform and doubling weights. These were either too weak or too aggressive. Values for λ_D are reported in Appendix B.1. As regularisation increases, characters are grouped into fewer vectors, eventually forming a single vector, similar to a sentence embedding. Over-regularisation causes the representation to collapse to the uninformative prior.

4.3 Evaluation of Abstract Representations

In this section we evaluate whether the model learns hierarchical and interpretable abstractions. We first analyse the learned representations qualitatively and quantitatively through attention visualisations and word segmentation (Section 4.3.1). We then probe the representations for linguistic information (Section 4.3.2) and evaluate them on a challenging sub-topic classification task (Section 4.3.2). Finally, we assess robustness to synthetic adversarial noise (Section 4.3.3).

Data All models are trained on the Wikitext-2 dataset (Merity et al., 2017) using a character-level tokeniser. The pretraining objective is to reconstruct the original text from noisy input, with random character deletions applied following Lewis et al. (2020).

Baseline models We compare our NVIB-regularised Transformer with a standard Transformer encoder. Both use the same architecture and training setup. For NVIB models, regularisation is applied only to the top three encoder layers. To ensure comparability, baseline models are trained with the same denoising objective and tuned to match the validation cross-entropy of the NVIB models on noised input (Appendix B.1).

Training details All models are trained from scratch using a consistent architecture. The encoder follows the six-layer base Transformer (Vaswani et al., 2017), and the decoder uses only two layers to limit its capacity and prevent it from compensating for weak encoder representations. All models use single-head attention, with embedding, projection, and feedforward dimensions set to 512, resulting in approximately 12–17 million parameters. Tokenisation is character-level, with a vocabulary of around 100 characters. Training runs for 55 epochs (about 8K steps), using the RAdam optimiser (Liu et al., 2020a) with mixed precision (FP16), a batch size of 512, gradient clipping of 0.1, and a learning rate of $1\mathrm{e}^{-3}$ with cosine decay. Dropout is set to 0.1. Each model trains in approximately 2.5 hours on a single NVIDIA GeForce RTX 3090.

NVIB hyperparameters NVIB introduces several components that require careful tuning. We use an exponential activation for the pseudo-count parameter $\boldsymbol{\alpha}$, and apply a threshold of 0.1 at test time to determine which vectors are dropped. To enforce a bottleneck, we mask final encoder outputs during training and testing using this same threshold. Hyperparameters λ_G, λ_D, and α^{Δ} are selected via tuning (Appendix B.1). We use a KL annealing schedule that linearly increases the regularisation strength between 30% and 60% of training steps. This allows the model to first learn meaningful representations, then gradually compress them. During training, we sample once per DP component, setting $\kappa^{\Delta} = 1$.

4.3.1 Interpretable Attention Maps

We evaluate abstraction by examining interpretability of the attention patterns. This is done qualitatively through visual inspection of attention maps and quantitatively through alignment with ground-truth word segments.

Visualising Abstraction To assess abstraction we visualise and interpret the self-attention maps. Figure 4.2 compares the final three encoder layers of two models: a baseline Transformer with six layers of standard attention (left), and a model with three standard layers followed by three NVIB-regularised denoising layers (right). Despite being trained only on noisy character-level reconstruction, the NVIB layers compress the representation into interpretable groups. Lower layers retain nearly all vectors (about

$\sim 99\%$) and attend locally, forming diagonal attention patterns. Higher layers drop many vectors (blank columns) and aggregate over longer spans (vertical bars), often aligning with subwords or full words. The final layer retains only around$\sim 35\%$ of vectors on average. This compression reflects stronger regularisation in higher layers, which encourages the model to group correlated characters and discard redundancy, most of which occurs within word boundaries. Additional examples are provided in Appendix B.3.

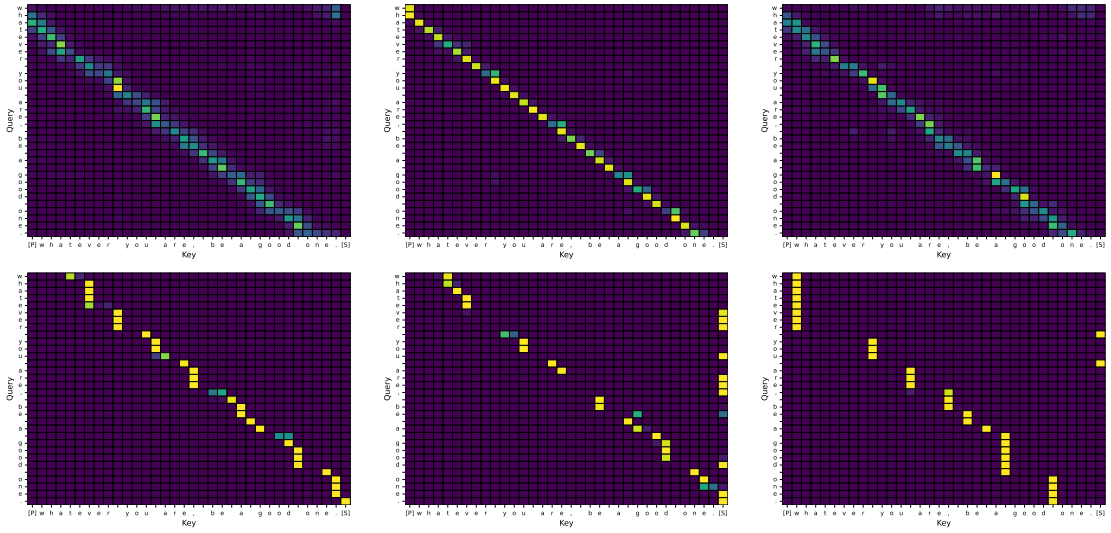


Figure 4.2: Self-attention patterns of the last 3 layers of 6-layer Transformer encoders from left to right. **Top**: Standard self-attention. **Bottom**: With NVIB regularisation. **Sentence**: "Whatever you are, be a good one." Dark purple is 0 and light yellow is 1 for attention.

Quantifying Abstraction We observe that NVIB models often segment around whitespace and word boundaries. To measure this, we evaluate how well final-layer attention segments align with ground-truth words. Contiguous segments are extracted by taking the $\arg\max$ over the Key dimension and grouping adjacent positions with the same target. For each segment-word pair, we compute the longest common substring and align the similarity matrix using the Hungarian algorithm (Kuhn, 1955). Precision (P), recall (R) are then defined as:

$$P = \frac{\text{longest common substring length}}{\text{segment length}}, \qquad R = \frac{\text{longest common substring length}}{\text{word length}}.$$

Table 4.1 reports macro-averaged results over the validation set. The baseline Transformer achieves high precision by capturing short within-word spans. In contrast, NVIB models recall more complete words and subwords, yielding a substantially higher F1 of 78.86%. This quantifies the emergence of interpretable, word-level abstractions seen in the attention maps.

Table 4.1: Word segmentation performance [%].

	P	R	F1
Transformer	**95.51**	56.51	64.52
NVIB	85.23	**79.02**	**78.86**

4.3.2 Probing for Linguistic Information

We probe the encoder to test whether its representations encode abstract linguistic properties. Two classifier types are used. The first is an attention-based probe that mimics how Transformers use information internally. It maps the sequence through a two-layer MLP, computes attention with a learnable query, and outputs a class prediction. The second is a simpler mean-aggregation probe, which averages the representations and feeds them into a two-layer MLP. These probes allow us to evaluate the abstractness and utility of different encoder layers.

SentEval Linguistic Probing We use SentEval to assess the linguistic information captured across all encoder layers (Conneau et al., 2018; Conneau and Kiela, 2018). This benchmark covers tasks from surface to semantic levels. We apply a mean-aggregation probe to both NVIB and baseline models, training for 10 epochs with a batch size of 128, a hidden dimension of 256, and the Adam optimiser with learning rate $1e{-}4$. All models are frozen during probing. We report test accuracy for the best validation checkpoint.

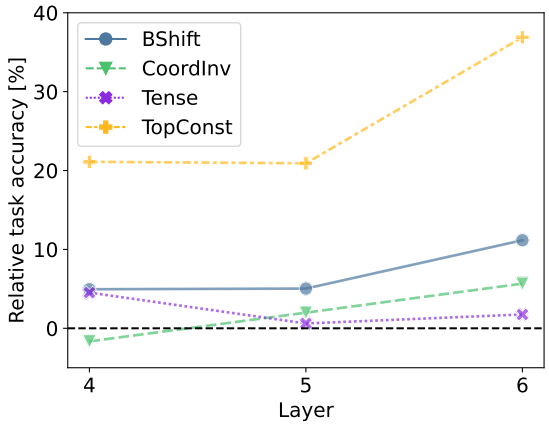


Figure 4.3: Relative performance improvement of NVIB over the baseline Transformer across the final three encoder layers on selected SentEval tasks. Higher scores indicate better linguistic abstraction. Tasks include syntactic (**BShift**, **TopConst**), semantic (**Tense**), and discourse-level (**CoordInv**) evaluations.

Figure 4.3 highlights four representative tasks. Success on **BShift** and **TopConst** requires capturing latent syntactic structure, while **Tense** and **CoordInv** require semantic and discourse-level understanding (Conneau et al., 2018). NVIB improves performance across layers, especially on deeper ones, indicating more abstract and linguistically informed representations. Full results are reported in Appendix Table B.2.

ArXiv Topic Classification We evaluate high-level abstraction using the ArXiv-L topic classification task (Hofmann et al., 2022), which involves assigning technical sentences to one of 20 sub-areas. Following Behjati and Henderson (2023), we train an attention-based probe on the final encoder layer. This task probes abstract linguistic properties without fine-tuning the underlying model (Hewitt et al., 2021). We use only the ArXiv-L subset, which contains $1K$ samples per class and $20K$ samples in total. The probe is a two-layer MLP with hidden size 256. Attention is computed using a learnable query, with key and value projections of size 512. The probe is trained for 50 epochs using the Adam optimiser with a learning rate of $1\mathrm{e}^{-3}$ and batch size 256. We report test F1 for the checkpoint with the highest validation F1.

Table 4.2: Test F1 score $[\%]$ on Arxiv-L classification task.

Task	Transformer	NVIB
Computer science	42.33	**44.47**
Mathematics	44.02	**47.13**
Physics	48.83	**52.32**
Average	45.06	**47.97**

Table 4.2 shows that NVIB improves classification performance across all domains. This suggests that the abstract representations capture more semantic information than the character-level units, providing more effective features for the task.

4.3.3 Robustness

We test model robustness to synthetic noise injected into the input sequence (Belinkov and Bisk, 2017; Durrani et al., 2019). We apply four types of perturbation: character swapping, deletion, insertion, and substitution (Morris et al., 2020). Robustness is measured by evaluating reconstruction quality under increasing noise levels. Figure 4.4 shows that NVIB models degrade slower than baseline Transformers. The performance gap widens as noise increases, suggesting that the compressed representations learned by NVIB are more stable and resilient.

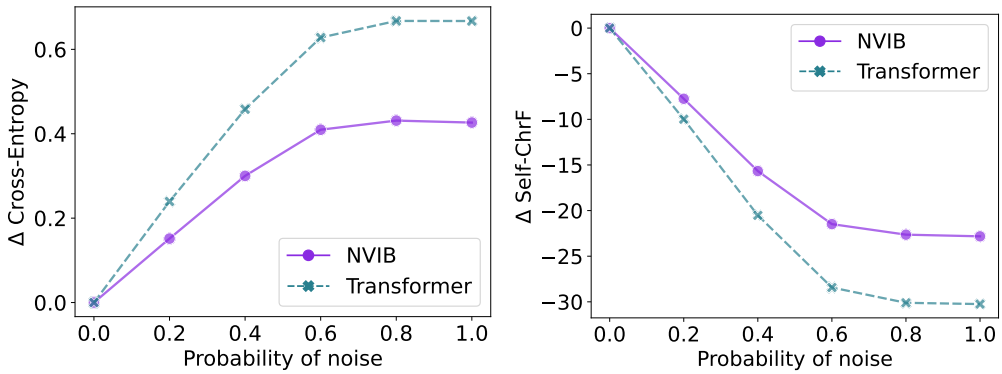


Figure 4.4: Robustness to synthetic character-level perturbations. Left: relative change in reconstruction cross-entropy. Right: change in ChrF score, a measure of character-level overlap. Lower degradation indicates better robustness.

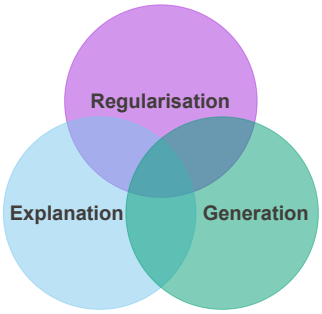
4.4 Conclusion

This chapter introduced a novel method for learning abstract and interpretable representations in Transformers. We adapted the Nonparametric Variational Information Bottleneck (NVIB) from Chapter 3 (Henderson and Fehr, 2023) to regularise self-attention in stacked encoder layers. This enables the model to learn how many vectors are needed at each layer, forming a hierarchy of abstraction within the model. We showed that these compressed representations are more interpretable, robust, and linguistically informative.

Our experiments used smaller models and character-level tokenisation on English data. Despite this, NVIB improves performance across tasks. We expect these gains to grow with larger models, morphologically rich languages, and alternative tokenisations that may benefit from compression. These findings suggest NVIB is well-suited for more practical and large-scale applications.

The next chapter investigates whether our latent variable model can characterise the information encoded by large pretrained Transformer representations. We apply NVIB to analyse these models through a Bayesian lens, aiming to uncover the inductive biases that support generalisation beyond the training distribution. We focus on machine translation, where abstraction may help handle morphological complexity, and summarisation, where compression is central to the task.

5 Pretrained Transformers with NVIB



Research Question

How do pretrained Transformer representations encode information and generalise out-of-distribution?

Chapter Summary

Pretrained Transformers have demonstrated impressive abilities but often fail to generalise out-of-distribution and are expensive to adapt to new domains. We explore how their representations can be better understood and improved through an information-theoretic lens. Extending Nonparametric Variational Information Bottleneck (NVIB), we reinterpret all attention functions in pretrained Transformers as instances of nonparametric variational models, using empirical priors and identity initialisation with tunable hyperparameters. We show that modifying these initialisations introduces a novel form of post-training regularisation that improves out-of-domain generalisation on translation and summarisation, without additional training. These findings suggest that the way information is encoded by pretrained Transformers is well-characterised by nonparametric variational Bayesian models.

Publication

- "Nonparametric Variational Regularisation of Pretrained Transformers",
  Fehr F., Henderson J.
  COLM 2024

Code Repository

The code is publically available at:

- https://github.com/idiap/nvib
- https://github.com/idiap/nvr_transformers.

Author Contributions

This paper was led by Fabio Fehr and supervised by James Henderson.

5.1 Generalising Beyond Pretraining

Self-supervised pretraining of transformer models (Devlin et al., 2019; Lewis et al., 2020; Raffel et al., 2020; Zhang et al., 2022b) has driven major advances across NLP. This indicates that the inductive bias of attention-based representations is extremely effective for language, though it remains unclear which characteristics of transformers are essential and which are incidental. We shed light on this issue by developing a variational Bayesian reinterpretation of pretrained transformers, which more explicitly characterises how they encode information about the data distribution. This insight enables improved out-of-distribution generalisation without any additional training.

In Chapter 3 Henderson and Fehr (2023) derive a variational Bayesian generalisation of a single-head cross-attention layer of transformers, called Nonparametric Variational Information Bottleneck (NVIB). In Chapter 4 Behjati et al. (2023) extended this framework to stacked self-attention layer for abstraction. When training from scratch, the NVIB layer provides an information-theoretic, sparsity-inducing regulariser over attention-based representations. In this Chapter, we investigate the possibility that NVIB also provides an accurate theoretical model of existing attention-based models which have been trained without NVIB regularisation. We extend NVIB to all the uses of attention in transformers (single- and multi- head; cross- and self- attention; in encoders and decoders), and propose a method for converting a pretrained transformer into the weights of an equivalent nonparametric variational (NV) Bayesian model (NV-Transformer).

This NV-Transformer is Bayesian in that it embeds text into a probability distribution over transformer embeddings, but remains equivalent by adding uncertainty around the embedding computed by the pretrained transformer, as illustrated in Figure 5.1. Theoretically, the exact equivalence only occurs when this uncertainty is exactly zero, but empirically we find a practical range of non-zero uncertainty levels where the model's predictions are unchanged (Section 5.3.2). Continuing training of this initial model with NVIB regularisation would push this uncertainty higher, but training large models is computationally expensive and requires significant amounts of in-domain data. Instead, we only adjust hyperparameters of the initialisation to better satisfy the NVIB regulariser by increasing uncertainty, without any backward-passes or parameter updates.

Increasing the uncertainty does alter model predictions, but without harming task accuracy. Instead, it acts as post-training regularisation, exploring alternative models with similar performance (Section 5.3.2). Surprisingly, this even improves out-of-domain generalisation (Sections 5.3.3 and 5.3.4). Small amounts of uncertainty remove unreliable features that do not generalise and can lead to overfitting. This agreement between the uncertainty in the Bayesian model and unreliability in pretrained transformer embeddings suggests our Bayesian reinterpretation helps us characterise how transformers encode information.

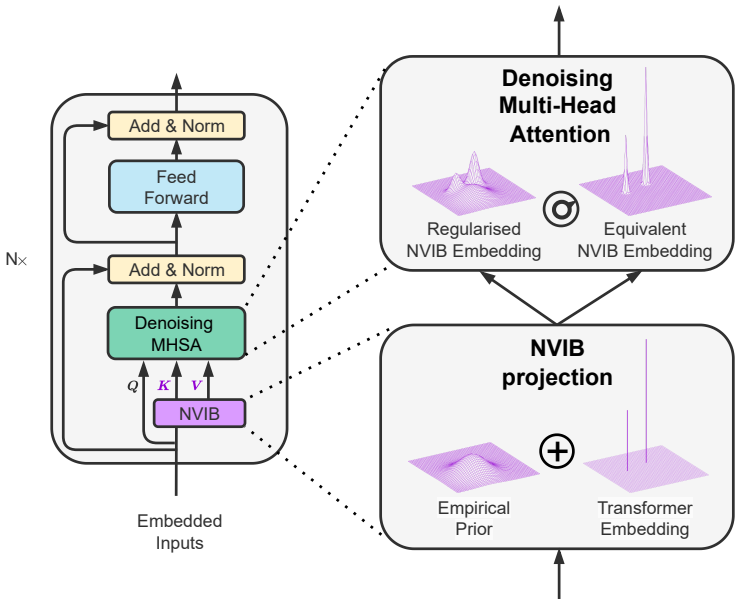


Figure 5.1: **Left:** Transformer encoder layer reinterpreted with NVIB and denoising multi-head attention (MHA). **Right:** NVIB projection layer combining transformer embeddings with an empirical prior, producing representations for denoising MHA. The dial controls the strength of post-training regularisation or allows for the equivalent nonparametric variational interpretation.

Contributions In this chapter we contribute technically to the development of novel models which use Nonparametric Variational Information Bottleneck regularisation, and to the understanding of pretrained transformers. **1.** We extend NVIB regularisation beyond single-head, cross-attention to all forms of multi-head attention in a transformer, resulting in our proposed NV-Transformer architecture (Section 5.2.1). **2.** We define a novel NVIB prior using the empirical distribution of a model's embeddings given a small amount of data (Section 5.2.3). **3.** We propose a reinterpretation of pretrained transformers as variational Bayesian models by defining a novel identity initialisation for NVIB with its controllable hyperparameters (Section 5.2.2). Adjusting these hyperparameters regularises the embeddings of this equivalent NV-Transformer without requiring retraining. **4.** Empirically, we show the usefulness of these proposals by smoothly varying the amount of this post-training regularisation (Section 5.3.2) and achieving improved performance in out-of-domain text summarisation (Section 5.3.3) and translation (Section 5.3.4). **5.** This successful Bayesian interpretation helps characterise the way that pretrained transformers represent reliable information in their embeddings (Section 5.4).

Related Work Related work in post-training regularisation such as quantisation (Dettmers et al., 2022; Yao et al., 2022; Xiao et al., 2023; Frantar et al., 2023b) and sparsity (Hubara et al., 2021; Frantar et al., 2023a, 2022) focus on regularising the model weights and not the latent embeddings. Similar to Frantar et al. (2023a), who propose a data-driven way to sparsify a model in one-shot without any retraining, we propose a data-driven form of soft sparsity for attention, without any retraining. Moreover, our method's regularisation uses information theory to regularise post-training. Our work shares similarity to the out-of-distribution generalisation literature in learning variational models with invariant features (Ilse et al., 2020) and parameter sharing in the embedding space to adapt to domain-shifts (Muandet et al., 2017; Li et al., 2017; Blanchard et al., 2021). We take a Bayesian approach and define our prior distribution based on empirical dataset statistics that allows for properties which are robust to domain shift. Park and Lee (2021) proposes a way to define pretrained language models as variational models by fine-tuning. Our work does not fine-tune or update the original model weights, nor does it train additional weights as in adapters (Houlsby et al., 2019; Hu et al., 2022a). We keep the model weights frozen and map attention layers to NVIB layers, setting just a few hyperparameters based on validation performance.

5.2 The NVIB Reinterpretation of Pretrained Transformers

We propose a novel method for reinterpreting pretrained transformers as nonparametric variational Bayesian models. To support this construction, we implemented the following applications of NVIB: multihead denoising attention; encoder denoising self-attention; and decoder denoising causal attention (Section 5.2.1). Pseudocode for denoising attention is given in Appendix C.4. This allows us to apply NVIB to every form of attention used in standard transformers, resulting in our proposed NV-Transformer reinterpretation of pretrained transformers. Using these extensions, we propose an identity initialisation with controllable hyperparameters, which allows us to achieve an equivalence with the standard attention mechanisms in transformers (Section 5.2.2). We then define a novel empirical prior distribution, which introduces uncertainty in the latent representations in a way that captures the implicit uncertainty in pretrained transformers (Section 5.2.3).

5.2.1 Denoising Multi-Head Attention

In this section, we extend the standard MHA formulation to the denoising setting. This builds on Section 2.1.2 and the practical denoising equations in Chapter 3, Equations 3.1 and 3.4. We derive the MHA form under denoising for both training and evaluation, covering both self and causal attention. This enables NVIB to be applied at every attention layer, yielding the NV-Transformer reinterpretation where NVIB is used throughout encoder and decoder blocks. The structure remains consistent with standard Transformers, but with information-theoretic regularisation added at each step (Figure 5.2).

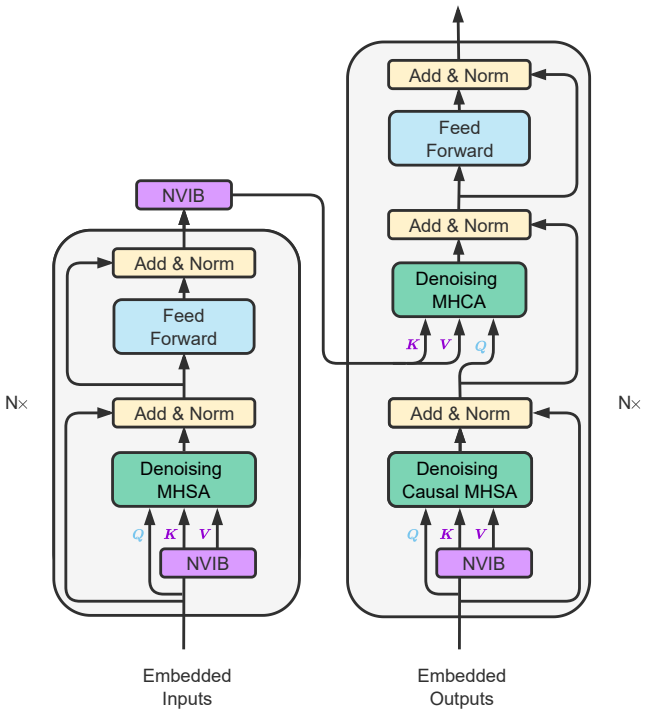


Figure 5.2: NV-Transformer: standard Transformer architecture reinterpreted with NVIB applied at every attention layer, including Denoising MHSA (multi-head self-attention), Denoising causal MHSA, and Denoising cross MHCA (multi-head cross-attention), across both encoder and decoder.

We define MHA using per-head attention scores. Let $\boldsymbol{U}' \in \mathbb{R}^{m\times d}$ be the queries and $\boldsymbol{Z} \in \mathbb{R}^{n\times d}$ the keys and values. The output is:

$$\mathrm{MHA}(\boldsymbol{U}', \boldsymbol{Z}) = \mathrm{Concat}_{i=1}^{h}\left(\mathrm{Softmax}\left(\boldsymbol{A}_i\right)\underbrace{\left(\boldsymbol{Z}\boldsymbol{W}_i^V + \boldsymbol{b}_i^V\right)}_{\boldsymbol{V}_i}\right),$$

$$\text{where } \boldsymbol{A}_i = \frac{1}{\sqrt{d/h}}\underbrace{\left(\boldsymbol{U}'\boldsymbol{W}_i^Q + \boldsymbol{b}_i^Q\right)}_{\boldsymbol{Q}_i}\underbrace{\left(\boldsymbol{Z}\boldsymbol{W}_i^K + \boldsymbol{b}_i^K\right)^\top}_{\boldsymbol{K}_i^\top} \in \mathbb{R}^{m\times n}.$$

Each head i uses a separate score matrix $\boldsymbol{A}_i$. The projections are split across heads. We define $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{h\times d\times \frac{d}{h}}$ and $\boldsymbol{b}^Q, \boldsymbol{b}^K, \boldsymbol{b}^V \in \mathbb{R}^{h\times \frac{d}{h}}$. This yields queries $\boldsymbol{Q} \in \mathbb{R}^{h\times m\times \frac{d}{h}}$ and keys $\boldsymbol{K} \in \mathbb{R}^{h\times n\times \frac{d}{h}}$ split over the h heads, as expected. We now expand the per-head scores before softmax.

$$\boldsymbol{A}_i = \frac{1}{\sqrt{d/h}}\left(\boldsymbol{Q}_i(\boldsymbol{Z}\boldsymbol{W}_i^K)^\top + \boldsymbol{Q}_i(\boldsymbol{b}_i^K)^\top\right) \tag{5.1}$$

The bias term $\boldsymbol{Q}_i(\boldsymbol{b}_i^K)^\top \in \mathbb{R}^{m\times 1}$ is constant across all n keys. It is cancelled by the softmax normalisation and typically omitted in practice. The scaling factor ensures stable gradients. It accounts for the reduced dimensionality d/h and controls the magnitude of dot products as d grows.

Denoising MHA during Training Although this chapter does not involve training, we include the training-time formulation of denoising MHA for completeness. We follow on from the practical formulation in Chapter 3, Equation 3.1. The NVIB layer outputs isotropic Gaussian parameters $\boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^{(n+1)\times d}$ and Dirichlet parameters $\boldsymbol{\alpha} \in \mathbb{R}^{(n+1)}$, where the $(n+1)^{\text{th}}$ component corresponds to the prior. During training, we sample $\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$ and $\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. The resulting attention function mirrors standard MHA with two modifications: (1) the keys come from the sampled vectors $\boldsymbol{Z} \in \mathbb{R}^{(n+1)\times d}$, which include a prior-sampled component; and (2) each key has a corresponding attention bias $\boldsymbol{b} \in \mathbb{R}^{(n+1)}$ defined by the sampled weights and norms. Concatenating over heads i, the denoising attention is

$$\text{DMHA}(.) = \text{Concat}_{i=1}^{h}\left(\text{Softmax}(\boldsymbol{A}_i + \underbrace{\log(\boldsymbol{\pi}) - \tfrac{1}{2\sqrt{d/h}}\|\boldsymbol{Z}\|^2)}_{\boldsymbol{b}}\ \underbrace{(\boldsymbol{Z}\boldsymbol{W}_i^V + \boldsymbol{b}_i^V)}_{\boldsymbol{V}_i}\right), \tag{5.2}$$

where $\boldsymbol{A}_i$ is defined in Equation 5.1. The bias term $\boldsymbol{b}$ consists of the log-probabilities $\log(\boldsymbol{\pi}) \in \mathbb{R}^{(n+1)}$ and a penalty based on the scaled L^2-norms of the sampled keys $\|\boldsymbol{Z}\|^2$. For multi-head attention, the extension is straightforward: we reuse the same $\boldsymbol{b}$ and $\boldsymbol{Z}$ across all heads.

Denoising MHA during Evaluation At evaluation time, we use a deterministic variant of denoising MHA, following the practical form in Chapter 3, Equation 3.4. The NVIB layer outputs isotropic Gaussian parameters $\boldsymbol{\mu}, \boldsymbol{\sigma} \in \mathbb{R}^{(n+1)\times d}$ and Dirichlet parameters $\boldsymbol{\alpha} \in \mathbb{R}^{(n+1)}$. Instead of sampling, we use these parameters directly, with three key differences from the training-time formulation: (1) the keys are deterministic, given by $\boldsymbol{\mu}$ scaled by the variance; (2) an attention bias $\boldsymbol{c} \in \mathbb{R}^{(n+1)}$ is added to each key; and (3) the output is a query-value interpolation weighted by posterior uncertainty. Letting $\boldsymbol{\sigma}_r^2 = \sqrt{d/h} + \boldsymbol{\sigma}^2$, the attention score matrix for each head i becomes

$$\boldsymbol{A}_i = \boldsymbol{Q}_i(\boldsymbol{W}_i^K)^\top \left(\frac{\boldsymbol{\mu}}{\boldsymbol{\sigma}_r^2}\right)^\top + \tfrac{1}{\sqrt{d/h}}\boldsymbol{Q}_i(\boldsymbol{b}_i^K)^\top.$$

As in training, the second term is broadcast across keys and normalised out. The key-wise bias is defined as

$$\boldsymbol{c} = \log\left(\frac{\boldsymbol{\alpha}}{\alpha_0}\right) - \frac{1}{2}\left\|\frac{\boldsymbol{\mu}}{\boldsymbol{\sigma}_r}\right\|^2 - \mathbf{1}_d(\log \boldsymbol{\sigma}_r)^\top, \tag{5.3}$$

where $\alpha_0 = \sum_j \alpha_j$ and $\mathbf{1}_d$ is a row vector of ones. This bias combines the log mixture weights, the scaled squared norms of the key vectors and the the influence of the variance.

The output of denoising attention is computed via an interpolation between the query and value vectors. Define $\boldsymbol{U}_i = \boldsymbol{Q}_i(\boldsymbol{W}_i^K)^\top \in \mathbb{R}^{m\times d}$ as the projected query. The interpolation weights, $\frac{\boldsymbol{\sigma}^2}{\boldsymbol{\sigma}_r^2}$ and $\frac{\sqrt{d/h}}{\boldsymbol{\sigma}_r^2}$, depend only on the values. The interaction matrix $\text{Softmax}(\boldsymbol{A}_i + \boldsymbol{c})$ typically propagates interaction information via the values alone. In the denoising setting, we factor the interpolation by applying the weights to the queries and values separately. This yields the following expression:

$$\begin{aligned}\text{DMHA}(.) = \text{Concat}_{i=1}^h \Bigg(&\left(\text{Softmax}(\boldsymbol{A}_i + \boldsymbol{c})\frac{\boldsymbol{\sigma}^2}{\boldsymbol{\sigma}_r^2}\right) \odot \boldsymbol{U}_i \\ &+ \text{Softmax}(\boldsymbol{A}_i + \boldsymbol{c})\left(\frac{\sqrt{d/h}}{\boldsymbol{\sigma}_r^2} \odot \boldsymbol{\mu}\right)\boldsymbol{W}_i^V + \boldsymbol{b}_i^V\Bigg)\end{aligned} \tag{5.4}$$

This formulation extends the single-head variant from Chapter 3, Henderson and Fehr (2023) to multi-head attention. It preserves the structure of standard attention while incorporating uncertainty-aware weighting and interpolation.

Denoising Self-Attention We apply denoising attention to encoder self-attention. This builds on Chapter 4, Behjati et al. (2023), where NVIB is used as a regulariser during training of single-head self-attention in stacked encoder layers. The queries are computed from the original n transformer vectors before projection to NVIB parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\alpha})$, while the keys and values are derived from the sampled vectors $\boldsymbol{Z} \in \mathbb{R}^{(n+1)\times d}$. This enables every instance of self-attention to use denoising attention. As in the prior work, we retain the exponential activation for pseudo-counts, but remove the skip connection between them, as it is not part of the pretrained transformer.

Denoising Causal Attention Causal attention is implemented using a mask in the decoder self-attention. This simulates next-token prediction with teacher-forcing (Vaswani et al., 2017). The mask is applied over the attention scores before the softmax. The prior component in the keys is never masked and acts as a fixed start-of-sequence token, without positional encoding. The attention bias $\boldsymbol{c}$ is unaffected by the mask, since the only mask-sensitive term is α_0, which normalises out. The implementation is therefore identical to denoising self-attention, with the addition of a diagonal mask.

5.2.2 Identity Initialisation for NVIB

We define an identity initialisation for NVIB such that denoising attention is effectively equivalent to standard attention. Given a set of vectors $\boldsymbol{Z}$ input to the attention layer, our proposed NVIB layer converts it to the parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\alpha})$ which specify a posterior DP distribution, and then applies the evaluation denoising attention function (Equation 5.4) to the resulting base distribution G_0^q. Excluding the prior component (discussed in Section 5.2.3), we define the projections:

$$\begin{aligned} \boldsymbol{\mu} &= \mu(\boldsymbol{Z}) = \boldsymbol{Z}\boldsymbol{W}^{\mu} + \boldsymbol{b}^{\mu} \\ \boldsymbol{\sigma}^2 &= \sigma^2(\boldsymbol{Z}) = \exp(\boldsymbol{Z}\boldsymbol{W}^{\sigma} + \boldsymbol{b}^{\sigma}) \\ \boldsymbol{\alpha} &= \alpha(\boldsymbol{Z}) = \exp(\boldsymbol{Z}^2\boldsymbol{w}_1^{\alpha} + \boldsymbol{Z}\boldsymbol{w}_2^{\alpha} + b^{\alpha}) \end{aligned} \tag{5.5}$$

where $\boldsymbol{Z}^2$ is the component-wise square. We choose these forms of projections because we want to set its parameters, $\boldsymbol{W}^{\mu}, \boldsymbol{W}^{\sigma} \in \mathbb{R}^{d\times d}$, $\boldsymbol{b}^{\mu}, \boldsymbol{b}^{\sigma} \in \mathbb{R}^d$, $\boldsymbol{w}_1^{\alpha}, \boldsymbol{w}_2^{\alpha} \in \mathbb{R}^d$ and $b^{\alpha} \in \mathbb{R}$, so that the evaluation denoising attention function is effectively equivalent to standard attention over $\boldsymbol{Z}$.

We obtain equivalence between denoising attention and standard attention through careful initialisation. This is achieved by matching Equation 5.4 to the standard attention mechanism, as visualised in Figure 5.1. To suppress the influence of the prior (the $n+1^{th}$ component), we ensure that the weights from the n data-dependent components dominate the prior component. This is done by setting the pseudo-counts $\alpha(\boldsymbol{Z})$ for the non-prior components to be large relative to the prior. In practice, this is achieved by setting the bias term b^{α} to a large positive value. To make the non-prior Gaussians approximate impulse distributions, we set the mean projection $\mu(\boldsymbol{Z})$ to the identity and initialise the variance projection $\sigma^2(\boldsymbol{Z})$ to output near-zero values. This removes uncertainty in both the attention weights and the query-value interpolation, effectively recovering the deterministic behaviour of standard attention.

To cancel the additive attention bias $\boldsymbol{c}$ in Equation 5.3, we require that it remains constant across keys. For this, the pseudo-count projection $\alpha(\boldsymbol{Z})$ must be proportional to the exponent of the scaled L^2 norm of $\boldsymbol{Z}$. This motivates the log-quadratic form of the projection in Equation 5.5, since it allows $\log\left(\frac{\boldsymbol{\alpha}}{\alpha_0}\right) \propto \frac{1}{2}\left\|\frac{\boldsymbol{\mu}}{\boldsymbol{\sigma}_r}\right\|^2$ thereby cancelling the corresponding term in $\boldsymbol{c}$ and preserving the attention weights.

Initialisation hyperparameters We define the parameters of these projections to achieve these requirements for equivalence but still allow some uncertainty. Based on preliminary experiments, we propose a small number of initialisation hyperparameters which controls the transition between equivalence and a smooth regularisation of the

embeddings (illustrated in Figure 5.1). The hyperparameters τ_α^i and τ_σ^i control the level of uncertainty of the mixture distributions by changing the initialisation of the pseudo-counts and variance, respectively. The indicator i is for the different sections of the model, which allows for independent control of the encoder's self-attention (e), decoder's cross-attention (c) and decoder's causal self-attention (d). Empirically, this allows for more flexibility than a single hyperparameter and is practically easier to tune than defining a hyperparameter per layer. In general we define the layer projection weights as follows:

$$\boldsymbol{W}^\mu = \boldsymbol{I}, \qquad \boldsymbol{b}^\mu = \mathbf{0}$$

$$\boldsymbol{W}^\sigma = \mathbf{0}, \qquad \boldsymbol{b}^\sigma = \log\left((\boldsymbol{\sigma}^p \cdot \tau_\sigma^i)^2\right) \tag{5.6}$$

$$\boldsymbol{w}_1^\alpha = \frac{1}{2\sqrt{d/h}} \odot \mathbf{1}, \qquad \boldsymbol{w}_2^\alpha = \mathbf{0}, \quad b^\alpha = \epsilon^\alpha \tau_\alpha^i \tag{5.7}$$

where d and h denote the model projection size and number of attention heads. As discussed in the next section, the empirical distribution is used to define the prior standard deviation $\boldsymbol{\sigma}^p$, and the constant ϵ^α, which denotes the empirical standard deviation of the scaled L_2^2-norm in log space per layer.

The hyperparameter τ_α^i in Equation 5.7 controls the relative weight of the prior component to the non-prior components in the mixture distribution G_0^q. Since ϵ^α is a standard deviation, it reflects the normal range of values for the non-prior-component pseudo-counts, so we can use τ_α^i to control the weight given to the prior component with respect to this range. When $\tau_\alpha^i = 0$, the non-prior pseudo-counts are their scaled L_2^2-norms and the prior pseudo-count is the expected scaled L_2^2-norm. When we increase or decrease the τ_α^i, it adjusts the magnitude of the non-prior L_2^2-norms, which relatively decreases or increases the weight on the prior proportionately to the standard deviation of the scaled L_2^2-norm.

The hyperparameter τ_σ in Equation 5.6 controls the interpolation between the query and value (Equation 5.4), which we set proportionately to the variance of the prior distribution. When $\tau_\sigma \approx 0$, there is effectively no interpolation, as with standard attention. When $\tau_\sigma = 1$, the uncertainty is increased to the level of the empirical prior distribution.

5.2.3 Empirical Prior of NVIB

The prior distribution is a Dirichlet Process, so it views all vectors in all transformer embeddings as impulses generated from its own base distribution. Therefore, we can estimate the base distribution of the prior empirically by observing the distribution of vectors given forward passes of the transformer model. Taking a Bayesian approach, the NV-Transformer's prior should reflect the distribution over vectors which the pretrained

transformer knows before seeing the input text. This is the distribution observed during training. We compute statistics from the fine-tuned corpora and use them to define our priors. We estimate the prior as the best fit of an isotropic Gaussian distribution to the empirical distribution over latent vectors computed when embedding this training corpus.

Empirical Prior Estimation We estimate prior parameters directly from the latent vectors $\boldsymbol{Z}$. An isotropic Gaussian prior $G_0^p \sim \mathcal{N}(\boldsymbol{\mu}^p, \boldsymbol{I}(\boldsymbol{\sigma}^p))$ is defined by:

$$\boldsymbol{\mu}^p = \frac{\sum_i^N \boldsymbol{Z}_i}{N},$$

$$(\boldsymbol{\sigma}^p)^2 = \frac{\sum_i^N (\boldsymbol{Z}_i - \boldsymbol{\mu}^p)^2}{N-1},$$

$$\log(\alpha_0^p) = \sum_i^N (\frac{\sum_j^d (\boldsymbol{Z}_{ij})^2}{2\sqrt{d/h}})/N,$$

where N is the total number of tokens in the training corpus, d is the dimension of the embedding and h is the number of attention heads. This allows the prior mean to be the least informative representation in the center of the latent embeddings vector space. The variance is estimated from this mean. The empirical pseudo-count is kept in log space for numerical stability and is the expected scaled L_2^2-norm of the latent vectors.

In Section 5.3.1, we analyse the empirical prior distributions across all encoder and decoder layers. We find that encoder self-attention, decoder causal self-attention, and decoder cross-attention each exhibit distinct embedding statistics. Within each group, the distributions are consistent. This motivates grouping the hyperparameters τ_α^i and τ_σ^i by attention type, where $i \in \{\text{encoder, cross, decoder}\}$.

5.3 Evaluation of Post-Training Regularisation

We evaluate our Bayesian reinterpretation of transformers through empirical analysis and controlled experiments. We begin by analysing the distribution of latent embeddings under the empirical prior, then show that identity initialisation yields empirical equivalence to pretrained transformers. A visualisation is shown in Figure 5.3 (left). This reinterpretation enables information-theoretic regularisation through modified NVIB initialisation. Once the empirical prior is estimated, adapting a pretrained model to a new domain reduces to hyperparameter selection using only forward passes. This is important as the cost of backward passes, regularisation, and parameter updates grows with model scale.

Our experimental design for Nonparametric Variational (NV) regularisation is entirely post-training and not directly comparable to fine-tuning methods. We first evaluate the empirical prior (Section 5.3.1) and identity initialisation formulations (Section 5.3.2) by assessing the in-domain performance on the original validation set. We then assess whether our regulariser improves out-of-domain (OOD) performance under domain shift for summarisation (Section 5.3.3) and translation (Section 5.3.4). This is increasingly relevant as models get larger, the cost of fine-tuning them on new data increases. We define the following setup: given a model θ_x^* trained on domain x, we evaluate performance on the same task in domain y, without further training. For instance, we test whether a news summarisation or translation model would generalises to informal dialogues. This demonstrates that simply modifying the initialisation of our model yields an information-theoretic form of post-training regularisation. A visualisation is shown in Figure 5.3 (right).

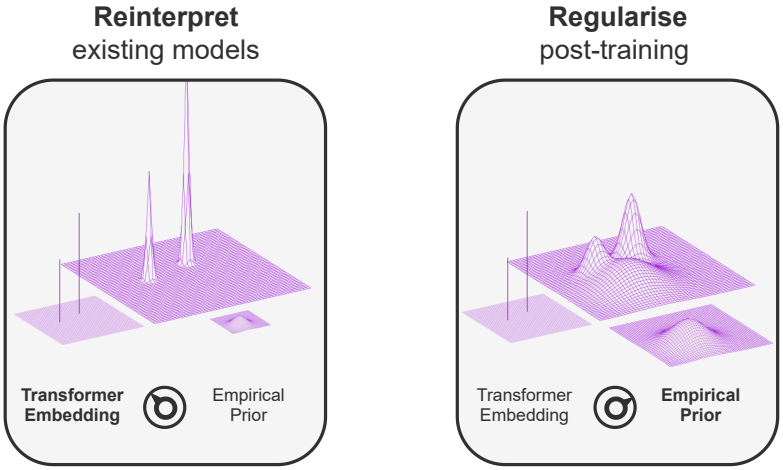


Figure 5.3: Embedding space visualisation of NVIB reinterpretation. **Left:** Reinterpretation with identity initialisation. Embeddings have low variance and are minimally influenced by the empirical prior. **Right:** Post-training regularisation via initialisation. Changing the initialisation to align with the empirical prior acts as a dial for controlling regularisation strength.

Data We use standard publically available summarisation and translation datasets. For summarisation, we include CNN/DailyMail (CNN/DM) (See et al., 2017), a widely used corpus of news articles with human-written highlights as summaries; Xsum (Narayan et al., 2018), where BBC articles are summarised by their first sentence. For OOD evaluation we use Curation Corpus (CC) (Curation, 2020), a set of professionally written summaries with a manual $50\%/25\%/25\%$ train-validation-test split; SAMsum (Gliwa et al., 2019), a dataset of chat-style dialogue summaries; and WikiHow, a corpus of instructional articles summarised by their headlines. For translation, we consider high-resource language pairs English-German (En-De) and English-French (En-Fr), using

OPUS100 (Zhang et al., 2020) as in-domain training data. Out-of-domain evaluation uses the Bible (Christodoulopoulos and Steedman, 2014), IWSLT 2017 (Cettolo et al., 2017), and TedTalks (Cettolo et al., 2012) datasets. For consistency, OOD corpora are split into train, validation, and test sets. TedTalks is evenly split; IWSLT is extended by reallocating 10K samples from training to validation; and the Bible corpus is split with 10K examples each for validation and test. Dataset statistics are summarised in the Appendix Tables C.1 and C.4.

Models We use pretrained and fine-tuned sequence-to-sequence transformer models, which are publically available (Wolf et al., 2020). For all models, we freeze the original weights and apply post-training regularisation by initialising the NVIB layers, which modify the attention mechanism without gradient updates. Empirical priors are constructed using the same corpora the models were fine-tuned on. For summarisation, we use BART (large) (Lewis et al., 2020), a denoising autoencoder with 12-layer encoder and decoder, 16 attention heads, 1024-dimensional embeddings, and 4096 feedforward dimensions. This results in approximately 406 million parameters. NVIB projections are added to each attention mechanism, increasing the parameter count by about 11% to 459 million. Input sequences are tokenised using a BPE tokenizer and truncated at 1024 tokens. Autoregressive generation uses beam search. For CNN/DailyMail, we set 4 beams, maximum length 142. For Xsum, we use 6 beams, maximum length 62. We evaluate BART models already fine-tuned on CNN/DailyMail [I] and Xsum. [II]. For translation, we use Marian (Junczys-Dowmunt et al., 2018), a transformer model with 6-layer encoder and decoder, 8 attention heads, 512-dimensional embeddings, and 2048 feedforward dimensions. This results in approximately 74 million parameters. Adding NVIB projections increases the total to 81 million parameters (about 9%). Input sequences are tokenised using SentencePiece and truncated at 512 tokens. Generation uses beam search with 4 beams and a maximum length of 512. We evaluate Marian models already fine-tuned on OPUS100 (Zhang et al., 2020) for English-German (En-De) [III] and English-French (En-Fr) [IV] pairs.

Baselines For baselines we consider the original model (32-bit) and quantisation of 16-bit and 8-bit (Dettmers et al., 2022). Quantisation is an alternative form of post-training regularisation which has been shown to improve performance (Dettmers and Zettlemoyer, 2022). We also considered a 4-bit quantisation model and combining NVIB regularisation with 16-bit quantisation (Appendix Table C.3, C.6). When combined, we found improvements over the original models. We did not consider sparsity techniques as they were not easily available without further training. In the following sections we consider sequence-to-sequence models trained on summarisation and machine translation.

[I] https://huggingface.co/facebook/bart-large-cnn
[II] https://huggingface.co/facebook/bart-large-xsum
[III] https://huggingface.co/Helsinki-NLP/opus-mt-en-de
[IV] https://huggingface.co/Helsinki-NLP/opus-mt-en-fr

NVIB hyperparameters The initialisation hyperparameters are defined in Section 5.2.2. We group them by attention type: encoder self-attention (e), cross-attention (c), and decoder causal attention (d). The parameter τ_α^i controls the weight of the prior component relative to the non-prior components in the mixture, for $i \in \{e, c, d\}$. The parameter τ_σ^i controls the interpolation between query and value, scaled proportionally to the standard deviation of the embeddings, where value of $\tau_\sigma^i = 1$ corresponds to one standard deviation. We perform random search over these hyperparameters using validation set forward passes only. Full details are in Appendix C.1 (summarisation) and C.1.2 (translation).

Evaluation metrics For summarisation, we report Rouge-1, Rouge-2, and Rouge-L (Lin, 2004), which measure unigram, bigram, and longest common subsequence overlap between generated and reference summaries. We also consider the summary word length as a proxy for information density. For translation, we use BLEU (Papineni et al., 2002), a precision-based n-gram overlap metric with a brevity penalty to discourage short outputs.

5.3.1 Empirical Prior Analysis

To further understand the distributions of our latent embeddings, we calculate the distribution of the empirical priors across all layers of the encoder and decoder attention mechanisms. Given a model θ_x^* that has been trained on data x, we analyse priors generated from both in-domain x and out-of-domain summarisation distributions. We estimate the empirical priors $(\boldsymbol{\mu}^p, \boldsymbol{\sigma}^p, \alpha_0^p)$ for a BART summarisation model (Lewis et al., 2020). Figure 5.4 shows average values per layer for each prior component for a model fine-tuned on Xsum (See Appendix C.3 for CNN/DailyMail). The final encoder layer (layer 13) corresponds to the cross-attention embedding used by the decoder.

We observe distinct patterns in the empirical prior across attention types. In the encoder, the mean is near zero and the variance is consistently low across datasets and models, except for the cross-attention layer where it drops. The expected log pseudo-count remains stable. In the decoder, mean activations are larger with similarly low variance, but log pseudo-counts increase sharply across layers, nearly doubling the encoder values in log-space. These results show stable distributions in the encoder and greater variability in the decoder. Encoder self-attention, decoder causal self-attention, and decoder cross-attention each show distinct statistics, but are consistent within each group. This motivates grouping the hyperparameters τ_α^i and τ_σ^i by attention type, where $i \in \{\text{encoder, cross, decoder}\}$.

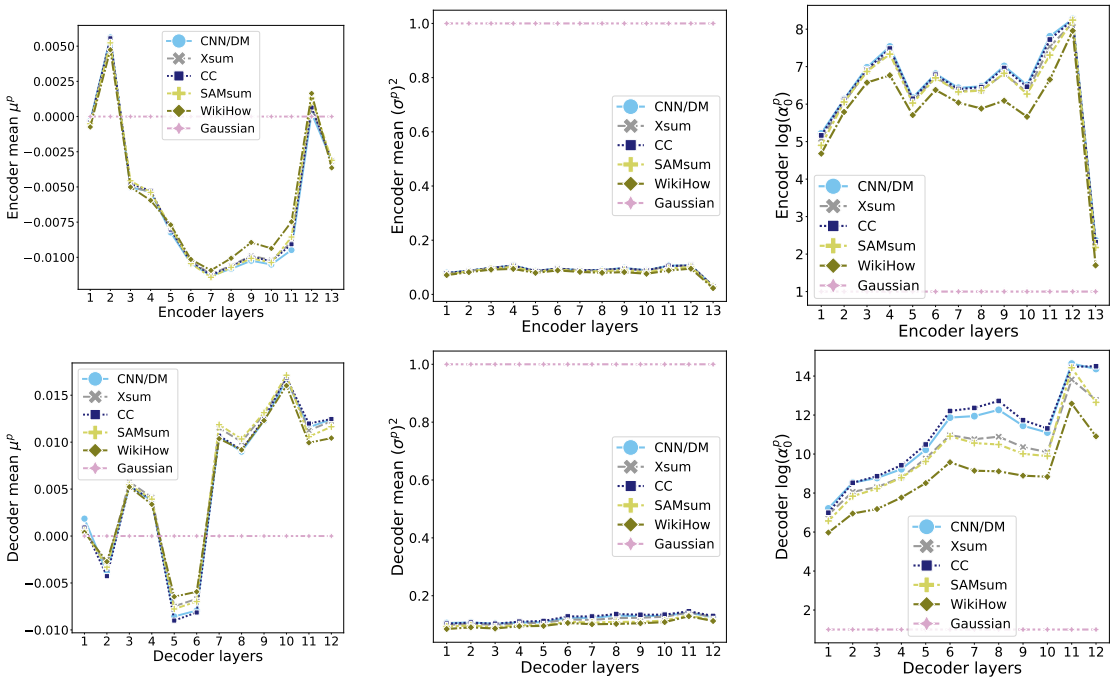


Figure 5.4: Distribution of embeddings for BART fine-tuned on Xsum. **Top**. Encoder **Bottom**: Decoder. Averaged empirical embeddings per layer of **Left:** mean component $\boldsymbol{\mu}^p$, **Middle:** variance $(\boldsymbol{\sigma}^p)^2$, **Right:** logged pseudo-count $\log(\alpha_0^p)$. "Gaussian" is a unit Gaussian, for reference.

Data Efficiency We assess the data efficiency of the empirical prior by subsampling the training set used to estimate its parameters. We evaluate Rouge-L on all models and datasets using the best validation-selected hyperparameters (Appendix C.1). Figure 5.5 shows that strong performance is maintained even when using as little as 0.1% of the training data (≈ 200 examples). This suggests that the empirical prior requires minimal data to be effective.

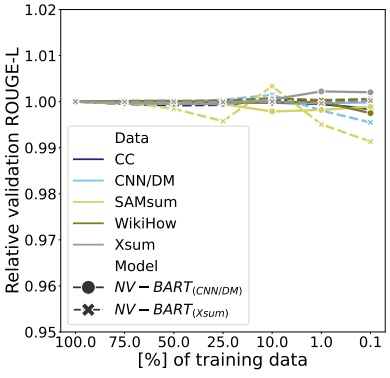


Figure 5.5: The relative validation performance of Rouge-L (y-axis) is compared for different amounts of training data used to create the empirical prior (x-axis).

5.3.2 Equivalent Identity Initialisation

We show that identity initialisation (Section 5.2.2) does not alter model predictions. This supports the hypothesis that pretrained transformers can be reinterpreted as NV models. We set the NVIB hyperparameters $\tau_\alpha^i = 10$ to be large and $\tau_\sigma^i \approx 0$ (10^{-38} for `Float32`) for all i, which down-weights the prior and suppresses query-value interpolation, as discussed in Section 5.2.2. Appendix Tables C.7 and C.8 confirms that the NV-Transformer achieves the same accuracy as the original model. Figure 5.6 shows that the leftmost points lie at 100% on the x-axis, indicating that both models generate identical outputs.

Analysis of Increasing Uncertainty We show that increasing the uncertainty within our regularisation allows access to models with different outputs but equivalent accuracy. We evaluate increasing the uncertainty of our NV models on the validation sets for the same datasets they were trained on. To characterise the range of these regularised NV models, Figure 5.6 plots them in a line ordered by their linear interpolations between the equivalent-initialisation and the everything-over-regularised corners of the sample space. The NV-Transformers are plotted by comparing their output using Rouge-L (left) or BLEU (right) against the gold summary or translation (y-axis) and against the original non-NV model's output (x-axis). As we increase the regularisation from the identity initialisation we discover a space of models which are not only different from the original baseline model (x-axis of $< 100\%$ overlap), but also equally good (same y-axis performance). The inclusion of our regularisation gives rise to a smooth transition between these models.

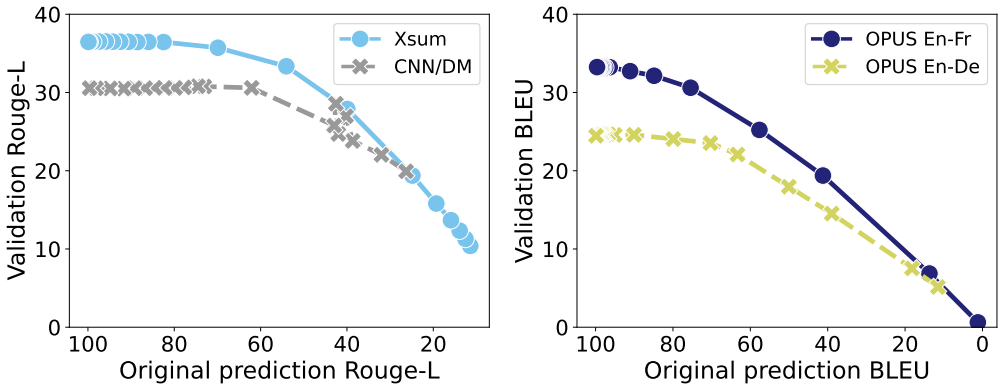


Figure 5.6: For each NVIB initialisation, the NV-Transformer's output is compared against the non-regularised original model (x-axis) and against the gold prediction (y-axis). Summarisation (left), Translation (right).

Attention maps We analyse attention maps from encoder, decoder causal, and decoder cross-attention across varying levels of NVIB regularisation (Appendix C.5). Visualisations are based on a Curation Corpus validation example, with selected layers and truncated input for clarity. Attention scores are averaged over heads and range from dark purple (0) to light yellow (1). Under identity initialisation ($\tau_\sigma^i \approx 0$, $\tau_\alpha^i = 10$), the model assigns no weight to the prior component $[P]$, despite its inclusion in the set of keys. With optimal regularisation (based on validation hyperparameters; see Appendix C.1), attention shifts from tokens such as punctuation to $[P]$. In an over-regularised setting ($\tau_\sigma^i \approx 0$, $\tau_\alpha^i = -30$), attention collapses entirely to $[P]$, reflecting a prior-dominated model. These patterns show that performance gains correlate with models that softly reweight attention toward the prior without full collapse.

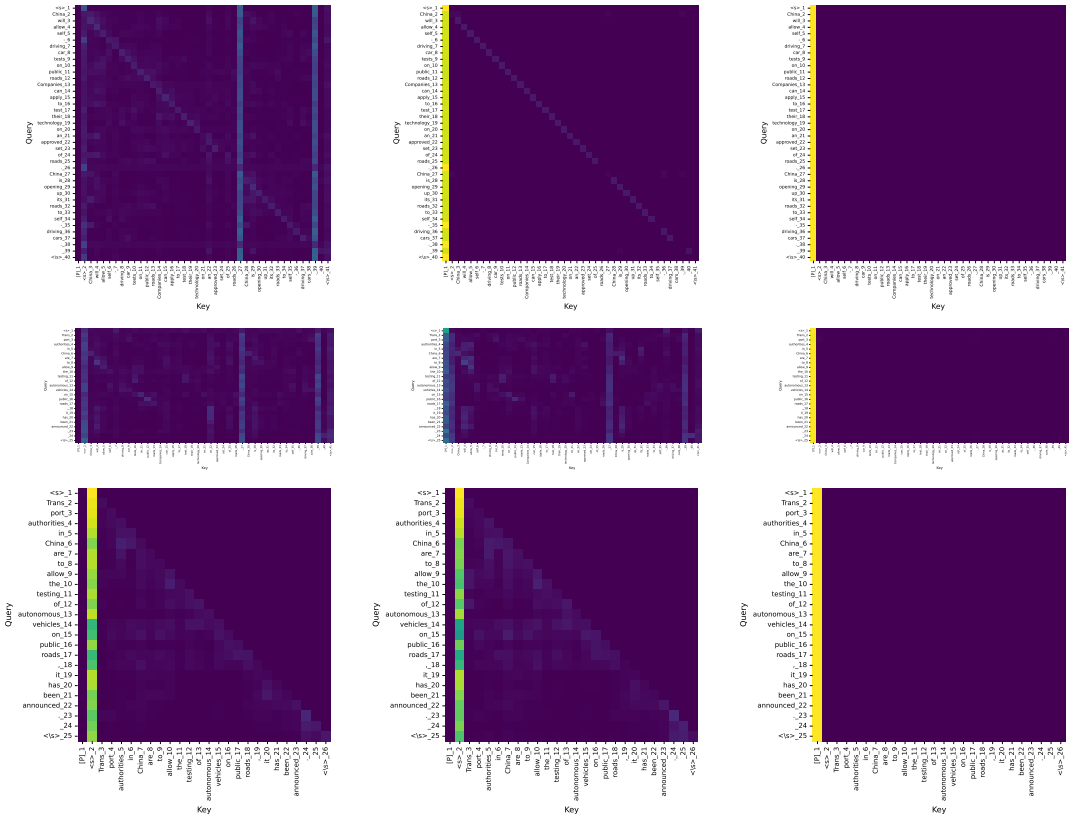


Figure 5.7: Attention maps from BART fine-tuned on XSum, using a short Curation Corpus example. Selected layers are shown, averaged over heads. Scores range from dark purple (0) to light yellow (1). **Top:** Encoder self-attention (layer 10). **Middle:** Decoder cross-attention (layer 3). **Bottom:** Decoder causal self-attention (layer 9). **Left:** Equivalence initialisation. **Center:** Best NVIB regularisation (validation selected). **Right:** Over-regularisation with prior collapse.

5.3.3 Summarisation Out-of-Domain Generalisation

We evaluate NV-Transformer models on out-of-domain summarisation tasks. Models are trained on either CNN/DailyMail or XSum and evaluated on the validation sets of three alternative summarisation datasets: Curation Corpus (CC) (Curation, 2020), SAMsum (Gliwa et al., 2019), and WikiHow (Koupaee and Wang, 2018). These datasets differ in style, structure, and content from the training data, enabling robust evaluation of domain generalisation. Further details on dataset statistics are provided in Appendix Table C.1. We report test-set results using the best-performing model selected on the corresponding validation set, as described in Appendix C.1. Table 5.1 reports the test set Rouge-L on the OOD text summarisation datasets for the post-training regularisation methods. Considering the original model trained on CNN/DailyMail, we notice that with NVIB regularisation the OOD performance on the test set improves over the baselines consistently, although minorly. In contrast, the original model trained on Xsum is substantially improved when NVIB regularisation is applied to OOD text summarisation. We speculate the larger improvement in the model trained on Xsum is due to the abstractive nature of the training data, whereas the model trained on CNN/DailyMail is more extractive. This shows that our information-theoretic, post-training regularisation can improve OOD generalisation, both over the original model and over quantisation.

Table 5.1: Post-training regularisation for out-of-domain (OOD) summarisation. We report test-set Rouge-L on both in-domain and OOD datasets. Values in brackets show the change relative to the base pretrained model.

			Out-of-Domain		
Model	**CNN/DM**	**Xsum**	**CC**	**SAMsum**	**WikiHow**
BART (CNN/DM)	**29.99**	13.12	24.99	22.42	9.26
BART-16bit (CNN/DM)	29.97 [-0.02]	13.13 [+0.01]	24.99 [0.00]	22.36 [-0.06]	9.27 [+0.01]
BART-8bit (CNN/DM)	29.55 [-0.44]	13.13 [+0.01]	24.67 [-0.32]	22.13 [-0.29]	9.36 [+0.10]
NV-BART (CNN/DM)	29.33 [-0.66]	**13.99 [+0.87]**	**25.04 [+0.05]**	**22.60 [+0.18]**	**9.41 [+0.25]**
BART (Xsum)	16.61	36.42	14.37	18.33	13.43
BART-16bit (Xsum)	16.60 [+0.01]	**36.44** [+0.02]	14.38 [+0.01]	18.24 [-0.09]	13.43 [0.00]
BART-8bit (Xsum)	16.56 [-0.05]	36.25 [-0.17]	14.33 [-0.04]	18.05 [-0.28]	13.53 [+0.10]
NV-BART (Xsum)	**19.42 [+2.81]**	36.25 [-0.17]	**17.61 [+3.24]**	**21.94 [+3.61]**	**15.25 [+1.82]**

Analysis of summarisation improvement We conjecture that the length of the summary is a strong proxy for the information within the document. Since the baseline model has been trained to produce a summary of a certain length, the summaries on OOD data can be improved by adapting the length of the summary to the information content.

We see this pattern in the mean sentence lengths in Table 5.2. The baseline produces summaries slightly shorter than its empirical Xsum training data for all datasets, while the reinterpreted NV-Transformer model matches this length in-domain but produces shorter summaries for shorter-summary datasets and longer summaries for longer-summary datasets.

Table 5.2: Mean number of words in validation set summaries for the original BART model trained on Xsum data and a reinterpreted NV-Transformer model.

Model	**CC**	**CNN/DM**	**SAMsum**	**WikiHow**	**Xsum**
Empirical	85.2	57.9	20.3	6.5	21.1
BART (Xsum)	19.3	20.7	17.4	16.4	18.9
NV-BART (Xsum)	28.2	30.8	16.6	14.4	18.6

Looking within the distribution of a single OOD dataset, we compare the output summaries of the original BART model and a reinterpreted NV-Transformer model on the OOD SAMsum test dataset. Figure 5.8 shows consistent improvements across the Rouge-1, Rouge-2 and Rouge-L distributions, which measures overlap metrics against the gold summaries. Figure 5.9 plots the output summary lengths against the true empirical summary lengths, and gives the Spearman's correlation coefficient. This suggests that the NV model is better at generalising to the length of the test summary according to the information variation in the input document.

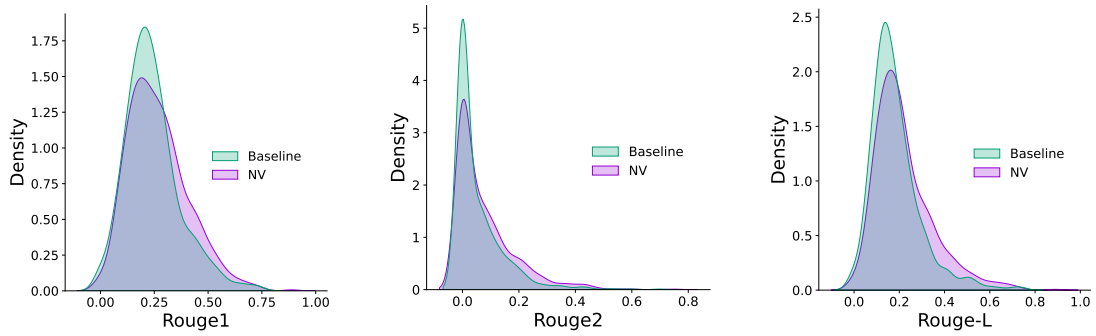


Figure 5.8: A BART baseline model trained on Xsum compared to its reinterpretation with NVIB regularisation on the SAMsum test dataset. **Left-Right**: Score distributions for Rouge-1, Rouge-2 and Rouge-L.

In Appendix Figure C.2 we provide these plots for all OOD validation datasets. We provide examples of generated summaries of all our summarisation models and in- and out-of-domain datasets in Appendix C.6.

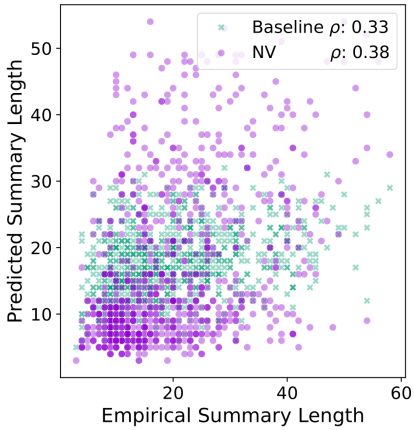


Figure 5.9: BART model trained on Xsum compared to its reinterpretation on the SAMsum test dataset. Empirical vs predicted summary length with Spearman's correlation.

5.3.4 Translation Out-of-Domain Generalisation

We evaluate translation models on English-to-German (En-De) and English-to-French (En-Fr) out-of-domain datasets. Specifically, we use the validation sets from Bible (Christodoulopoulos and Steedman, 2014), IWSLT (Cettolo et al., 2017), and TedTalks (Cettolo et al., 2012). Appendix Table C.4 provides dataset statistics. We select the best model on the validation set (Appendix C.6) and report its performance on the test set. Table 5.3 shows BLEU scores on the OOD test sets.

Table 5.3: Post-training regularisation for out-of-domain (OOD) translation. We report test-set BLEU on both in-domain and OOD datasets. Values in brackets show the change relative to the base pretrained model.

		Out-of-Domain		
Model	**OPUS100**	**Bible**	**IWSLT**	**TedTalks**
Marian (OPUS En-De)	24.78	23.02	27.16	24.80
Marian-16bit (OPUS En-De)	24.70 [-0.08]	23.02 [0.00]	27.18 [+0.02]	24.79 [-0.01]
Marian-8bit (OPUS En-De)	24.65 [-0.13]	**23.11 [+0.09]**	27.05 [-0.11]	24.67 [-0.13]
NV-Marian (OPUS En-De)	**24.84 [+0.06]**	22.95 [-0.07]	**27.30 [+0.14]**	**25.06 [+0.26]**
Marian (OPUS En-Fr)	35.22	27.23	39.03	31.92
Marian-16bit (OPUS En-Fr)	35.22 [0.00]	27.21 [-0.02]	39.05 [+0.02]	31.97 [+0.05]
Marian-8bit (OPUS En-Fr)	**35.23 [+0.01]**	27.19 [-0.04]	39.03 [0.00]	31.89 [-0.03]
NV-Marian (OPUS En-Fr)	35.22 [0.00]	**27.41 [+0.18]**	**39.28 [+0.25]**	**32.43 [+0.51]**

We observe consistent but modest improvements in 5 out of 6 cases. We hypothesise that translation, as a direct mapping between languages, benefits less from information reduction compared to summarisation. These results further support our hypothesis that post-training, information-theoretic regularisation improves OOD generalisation of pretrained transformers.

5.4 Discussion

Since regularisation does not guarantee improved performance, it is surprising that our method, which is applied post-training without backpropagation or parameter updates, can improve performance. We believe the variance in performance is attributed to the nature of the different tasks. Summarisation is fundamentally an information compression task, well suited to an information bottleneck, unlike translation. We speculate the larger improvement in summarisation models trained on Xsum is due to the abstractive nature of the dataset. The Xsum summarisation dataset requires more generalisation than the extractive CNN/DailyMail dataset, which relies on selecting a summary from the document.

The improvements from post-training regularisation suggest that our variational Bayesian reinterpretation of pretrained transformers is an accurate model of how information is being captured in transformer embeddings. When we map an embedding into a probability distribution, the variance specifies what differences between embeddings constitute reliable information, and what differences should be unreliable or unimportant. The denoising attention function then ignores this unreliable information, which should not affect accuracy in-domain, and could reduce overfitting out-of-domain. The above experiments succeed in finding such empirically effective patterns of variance for our proposed DP distributions (Figures 5.6, 5.7 and Tables 5.1, 5.3). Since this regularisation is done post-training, this argument suggests that the pretrained model has learned what information is unreliable, and that NVIB regularisation exposes this representation of information. In particular, our model suggests that the information in a given embedding dimension value is relative to the distribution over that dimension during training, with the mean value carrying no information and the scale being relative to the variance. They also suggest that the L_2^2-norm of a vector reflects the importance of the information in a vector. We anticipate that these insights into how transformer embeddings represent information will help in the understanding and development of future improvements to transformer architectures.

5.5 Conclusion

This chapter provides insight into how pretrained transformers encode information and generalise out-of-distribution. We contribute both to the development of Nonparametric Variational Information Bottleneck (NVIB) models and to the understanding of pretrained transformers. Technically, we extend NVIB to all forms of multi-head attention, introduce a novel empirical prior, and propose an effective identity initialisation with interpretable hyperparameters. These yield the NV-Transformer: a denoising-attention model that behaves identically to a pretrained transformer under identity initialisation.

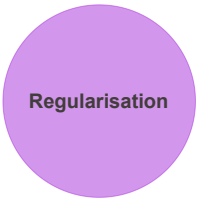
We show that pretrained transformers encode information in a way that is well modelled by nonparametric variational distributions. By increasing uncertainty from identity initialisation, we uncover a smooth space of models that differ from the original yet perform equally well. The denoising attention mechanism down-weights unreliable or unimportant information in the embeddings. This allows the NVIB regularisation to reveal how pretrained transformers implicitly distinguish between informative and uninformative latent representations.

We evaluate this reinterpretation through empirical analysis and post-training experiments in summarisation and translation. Despite being applied without backpropagation or parameter updates, NVIB improves out-of-domain generalisation in most settings. This suggests that pretrained transformers already encode generalisable structure, and that NVIB regularisation makes this structure explicit. In particular, the effectiveness of NVIB on Xsum—a more abstractive summarisation task—compared to CNN/DailyMail highlights how task properties interact with information bottleneck principles. These results support our hypothesis that pretrained representations encode uncertainty, and that controlling this uncertainty can reduce overfitting out-of-domain.

We foresee NVIB regularisation being especially valuable for decoder-only language models, where backward passes are expensive and post-training control is desirable. For large-scale models, however, it is non-trivial to integrate NVIB with relative positional encodings (Su et al., 2024) and hardware-specific implementations such as flash-attention (Dao et al., 2022). In this chapter, we restrict our focus to encoder-decoder transformers, as they include all three core attention types: encoder self-attention, decoder causal attention, and cross-attention.

So far, we have evaluated NVIB regularisation only in post-training settings and only for NLP tasks. Real-world application often requires adaptation during fine-tuning and across diverse modalities. This raises a broader question: how can generalisable representations be induced during fine-tuning across different models and modalities? We address this in the next chapter.

6 Fine-tuning Transformers with NVIB



Research Question

How can generalisable representations be induced during fine-tuning across different models and modalities?

Chapter Summary

Pretrained attention-based models often struggle with generalisation, particularly under distribution shifts, out-of-domain transfer, and few-shot settings. This limitation spans modalities such as speech, text, graphs, and vision. Nonparametric Variational Information Bottleneck (NVIB) is an attention-based regulariser that improves generalisation, but has so far only been applied to text and without fine-tuning. We investigate whether NVIB's ability to remove information from pretrained embeddings reduces reliance on spurious features during fine-tuning. We are the first to integrate NVIB regularisation into fine-tuning across diverse models and modalities. This required architectural changes to allow for adaptability, improve stability, and simplify evaluation. We observe improved out-of-distribution generalisation in speech quality assessment and language identification, text with induced attention sparsity, graph-based link prediction, and image-based tasks, including few-shot classification and privacy classification.

Publication

- "Fine-Tuning Pretrained Models with NVIB for Improved Generalisation"
  Fehr F., Baia A. E.*, Chang X.*, Coman A. C.*, El Hajal K.*, El Zein D.*, Kumar S.*, Zuluaga-Gomez J. P.*, Cavallaro A., Teney D., Henderson J.,
  ICLR 2025 Workshop on Spurious Correlation and Shortcut Learning

Code Repository

The code is publically available at:

- https://github.com/idiap/nvib
- https://github.com/idiap/nvib_finetuning.

Author Contributions

Fabio Fehr led the large collaboration and implemented NVIB across all models. PhD contributors*, who contributed equally, conducted modality-specific experiments under the supervision of Andrea Cavallaro, Damien Teney, and James Henderson.

6.1 Learning Generalisable Representations

In this chapter we investigate how generalisable representations can be induced during fine-tuning. This extends the reinterpretation of pretrained transformers from Chapter 5 (Fehr and Henderson, 2024). Leveraging pretrained attention-based representations by fine-tuning has become the de facto modelling paradigm due to its wide applicability and significant improvements on the state-of-the-art (Ruder et al., 2019). Applications of pretrained Transformers (Vaswani et al., 2017) are modality agnostic and gained prevalence across: speech processing (Baevski et al., 2020; Radford et al., 2023); natural language processing (Devlin et al., 2019; Raffel et al., 2020; Touvron et al., 2023), graphs Rong et al. (2020); Li et al. (2021b) and computer vision (Liu et al., 2021; Dosovitskiy et al., 2021; Bao et al., 2022).

The success of pretrained attention-based models is thought to stem from their ability to scale, both in terms of corpora size and the number of parameters, as well as the inductive biases inherent in the attention-based architecture (Henderson, 2020; Zhai et al., 2022; Fedus et al., 2021; Dehghani et al., 2023). Despite their success, these models still exhibit notable limitations during fine-tuning. Due to their large number of parameters and expressivity, they can be prone to overfitting and struggle to generalise in the presence of shortcuts from spurious correlations (Bhargava et al., 2021; Geirhos et al., 2020), distribution shift (Wu et al., 2020a; Kumar et al., 2022). The attention mechanism facilitates expressivity through token interaction, but this also introduces redundant information, which can hinder generalisation (Bian et al., 2021; Bhojanapalli et al., 2021). A well-established strategy to combat overfitting is to inject noise during training, which improves generalisation by encouraging robustness (Ferianc et al., 2023). This principle underlies regularisation techniques such as Dropout, which have proven effective across vision, speech, and text (Srivastava et al., 2014). Similarly, sparsity-based regularisation in attention has been shown to reduce redundancy and improve generalisation (Child et al., 2019; Behjati et al., 2023; Fehr and Henderson, 2024). However, integrating such regularisation into the fine-tuning of pretrained models remains both challenging and largely unexplored.

Information Bottleneck (IB) methods offer an information-theoretic approach to noise-based regularisation, shown to reduce generalisation error. As explained in Section 2.2.1, the IB principle learns latent features Z that compress the input X while retaining information relevant to the output Y (Tishby et al., 2000). The variational information bottleneck (VIB) framework, introduced through a variational lower bound to the IB objective (Alemi et al., 2017), enables deep neural representations (Tishby and Zaslavsky, 2015) to be trained using gradient-based optimisation. This framework has been widely applied across speech (Nelus and Martin, 2021; Lian et al., 2022), natural language (McCarthy et al., 2020; mahabadi et al., 2021), graphs (Wu et al., 2020b; Sun et al., 2022) and vision (Han et al., 2020; Chun, 2024). The success of the VIB framework can be attributed to its key properties, including resilience against spurious correlations

(Chuah et al., 2022) and distribution shift (Li et al., 2021a), robustness (Zhang et al., 2022a) and sparsity (Paranjape et al., 2020). Despite this success, VIB regularisation has seen limited exploration in the fine-tuning of pretrained attention-based models. Applying VIB to these pretrained models is difficult due to the complexity of incorporating it into the variable-sized latent representations accessed by attention.

In Chapter 3, Henderson and Fehr (2023) propose the Nonparametric Variational Information Bottleneck (NVIB) as a VIB regulariser for attention layers. NVIB is uniquely suited to regularise the variable-sized representations accessed by attention, as it compresses both the information within individual vectors and the number of vectors. Subsequent contributions demonstrate that NVIB exhibits desirable properties such as out-of-distribution (OOD) generalisation, robustness, and sparsity, as shown in Chapters 3 (Henderson and Fehr, 2023), 4 (Behjati et al., 2023), and 5 (Fehr and Henderson, 2024). In Chapter 4, NVIB is applied for representation learning by integrating the regulariser into the self-attention layers of a Transformer encoder, trained from scratch to learn progressively sparser representations. Chapter 5 extends NVIB to pretrained models, reinterpreting them as Bayesian models and achieving improved OOD performance in summarisation and translation without further training. This chapter continues naturally by applying NVIB regularisation during fine-tuning of pretrained models. We further explore how nonparametric variational models can generalise beyond text to diverse modalities such as vision, speech, and graphs, each with their own model architectures, data types, and tasks.

Contributions. In this chapter, we extend NVIB regularisation methods to fine-tuning, with diverse pretrained models. **1.** We propose several novel methods for NVIB fine-tuning, including a simplified denoising attention function at evaluation (Section 6.2.1), learnable prior mean embedding per layer for adaptability (Section 6.2.2), and a clipped Dirichlet pseudo-counts for stability (Section 6.2.3). **2.** We do the first empirical evaluation of NVIB on diverse modalities such as speech (Section 6.3.1), text (Section 6.3.2), graphs (Section 6.3.3), and vision (Section 6.3.4 and 6.3.5). **3.** We show improved OOD generalisation in classification and regression tasks, demonstrating NVIB's added value across diverse applications.

6.2 Fine-Tuning with NVIB

This section introduces our methodology for incorporating NVIB regularisation during fine-tuning of pretrained models. Firstly, we simplify the denoising function at evaluation by aligning it more closely with the training-time function (Section 6.2.1). We then introduce a learnable prior by relaxing the fixed prior assumption from Chapter 5, allowing the model to better adapt to the pretrained representations and downstream task (Section 6.2.2). To improve stability during optimisation, we apply proportional clipping

to the Dirichlet concentration parameters $\boldsymbol{\alpha}$, which mitigates the numerical instability introduced by the initialisation (Section 6.2.3). Finally, we define a regularised fine-tuning loss that combines the task-specific objective with KL terms from the NVIB layer to support both task performance and out-of-distribution generalisation (Section 6.2.4).

Before fine-tuning, we start by reinterpreting the pretrained models as nonparametric variational models by inserting NVIB layers before each attention mechanism, as shown in Figure 6.1. Each NVIB layer maps the input $\boldsymbol{x}$ to Dirichlet Process (DP) parameters $(\boldsymbol{\mu}^q, \boldsymbol{\sigma}^q, \boldsymbol{\alpha}^q)$ and is initialised to approximate the identity, following the procedure in Section 5.2.2. All parameters, including those of the NVIB layer, are updated during fine-tuning.

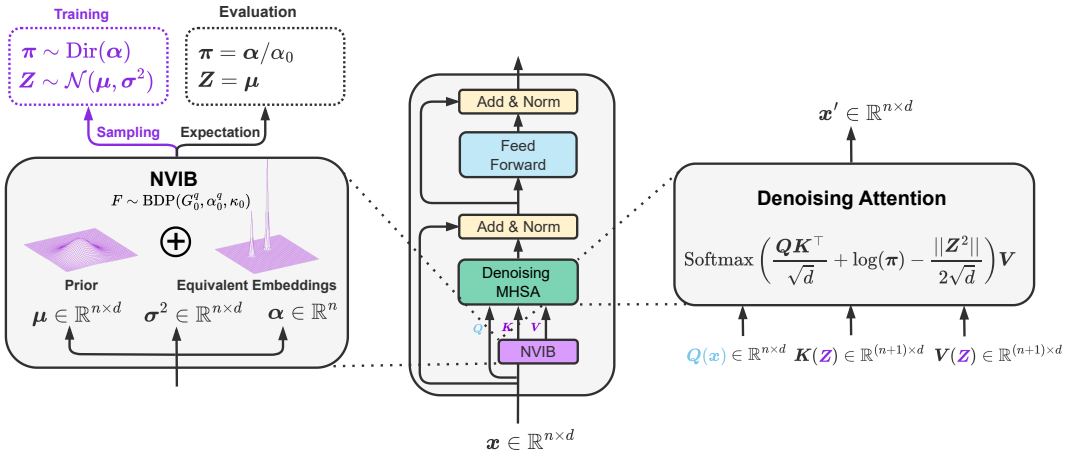


Figure 6.1: **Center:** Transformer encoder layer reinterpreted with NVIB and denoising multi-head attention (MHA). **Left:** NVIB layer with identity initialisation. **Right:** Simplified denoising MHA illustrating the alignment between training and evaluation-time functions.

6.2.1 Simplifying Denoising Attention

During fine-tuning, it is important that the attention function remains consistent between training and inference. To this end, we introduce a simplified denoising formulation.

Recall that the NVIB layer projects the sequence of vectors $\boldsymbol{x} \in \mathbb{R}^{n \times d}$ from a Transformer embedding to the parameters of a Dirichlet Process. These include isotropic Gaussian means $\boldsymbol{\mu} \in \mathbb{R}^{n \times d}$ and variances $\boldsymbol{\sigma}^2 \in \mathbb{R}^{n \times d}$ defining the mixture base distribution, and Dirichlet concentration parameters $\boldsymbol{\alpha} \in \mathbb{R}^n$. Each of the n input vectors maps to a mixture component, with an additional $(n{+}1)^{\text{th}}$ component serving as a prior.

Training-Time Denoising Attention During training, the NVIB layer samples a mixture distribution as a set of weighted vectors $(\boldsymbol{\pi}, \boldsymbol{Z})$, where $\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$ and each $\boldsymbol{Z}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$. The sampled vectors $\boldsymbol{Z} \in \mathbb{R}^{(n+1)\times d}$ serve as the keys in the attention mechanism, including a sample from the prior component. Each key receives a bias $\boldsymbol{b} \in \mathbb{R}^{(n+1)}$ computed from its corresponding weight $\boldsymbol{\pi} \in \mathbb{R}^{(n+1)}$. We present the single-head case here, omitting the head index, which generalises to multi-head attention as described in Section 5.2.1 and Equation 5.2. The denoising attention function is defined as:

$$\text{DAttention}(.) = \text{Softmax}\left(\boldsymbol{A} + \underbrace{\log(\boldsymbol{\pi}) - \tfrac{1}{2\sqrt{d}}\|\boldsymbol{Z}\|^2}_{\boldsymbol{b}}\right)\underbrace{(\boldsymbol{Z}\boldsymbol{W}^V + \boldsymbol{b}^V)}_{\boldsymbol{V}}.$$

The bias $\boldsymbol{b}$ combines $\log(\boldsymbol{\pi})$ and the scaled squared L2 norms of $\boldsymbol{Z}$. The same $\boldsymbol{Z}$ and $\boldsymbol{b}$ are reused across all heads.

Simplified Inference-Time Denoising Attention During evaluation, the NVIB layer outputs expected mixture parameters instead of sampling. We approximate this by setting $\boldsymbol{Z} = \boldsymbol{\mu}$ and $\boldsymbol{\pi} = \boldsymbol{\alpha}/\sum_i \alpha_i$, where we define $\alpha_0 = \sum_i \alpha_i$ for convenience. This removes training-time stochasticity while preserving the structure of the learned mixture. As in training, the NVIB layer outputs the parameters $\boldsymbol{\mu} \in \mathbb{R}^{(n+1)\times d}$, $\boldsymbol{\sigma} \in \mathbb{R}^{(n+1)\times d}$, and $\boldsymbol{\alpha} \in \mathbb{R}^{(n+1)}$, but we use the mean base distribution directly. The attention scores $\boldsymbol{A} \in \mathbb{R}^{m\times(n+1)}$ are computed as:

$$\boldsymbol{A} = \boldsymbol{Q}(\boldsymbol{W}^K)^\top\left(\frac{\boldsymbol{\mu}}{\sqrt{d}}\right)^\top + \frac{1}{\sqrt{d}}\boldsymbol{Q}(\boldsymbol{b}^K)^\top.$$

The term $\boldsymbol{Q}(\boldsymbol{b}^K)^\top$ is constant across all keys and cancels out in the softmax. Following the formulation in Equation 5.4, the simplified evaluation-time denoising attention function is:

$$\text{DAttention}(.) = \text{Softmax}\left(\boldsymbol{A} + \underbrace{\log(\tfrac{\boldsymbol{\alpha}}{\alpha_0}) - \tfrac{1}{2\sqrt{d}}\|\boldsymbol{\mu}\|^2}_{\boldsymbol{c}}\right)\underbrace{(\boldsymbol{\mu}\boldsymbol{W}^V + \boldsymbol{b}^V)}_{\boldsymbol{V}}$$

The bias $\boldsymbol{c}$ matches the training-time bias $\boldsymbol{b}$ but uses expected values, combining $\log(\frac{\boldsymbol{\alpha}}{\alpha_0})$ with the scaled squared L2 norms of $\boldsymbol{\mu}$. This formulation simplifies previous versions by removing dependence on variance in the mean and omitting the interpolation between query and value vectors. It improves alignment between training and evaluation, simplifies implementation, and reduces computational cost. Empirical ablations in Table D.4 demonstrate the benefits of this simplification. Full pseudocode is provided in Appendix D.3.

6.2.2 Learnable Prior

We introduce a learnable prior mean to improve flexibility during fine-tuning. The intuition is that the prior should represent an uninformed center of the embedding space. In contrast to Chapter 5 (Fehr and Henderson, 2024), which estimates the prior from training data, we initialise the prior mean $\boldsymbol{\mu}^p = \mathbf{0}$ and allow it to be learned. This enables adaptation to both the pretrained model and the downstream task. To preserve regularisation, we keep the prior variance fixed at $(\boldsymbol{\sigma}^p)^2 = \mathbf{1}$ and the pseudo-count at $\alpha_0^p = 1$.

Appendix Table D.4 shows that the learnable prior has minimal impact on performance. This aligns with the empirical finding in Section 5.3.1 that encoder embeddings are already centred near zero. While zero is a reasonable initialisation, learning the prior offers flexibility. We hypothesise that this may become increasingly beneficial as model size and complexity grow, and leave further investigation to future work.

6.2.3 Dirichlet Parameters Clipping

To match the behaviour of the pretrained model, the identity initialisation requires large values of $\boldsymbol{\alpha}$. While this was stable in training from scratch in Chapter 3, it causes numerical instability during fine-tuning. We address this by applying proportional clipping to the Dirichlet concentration parameters. The magnitude of $\boldsymbol{\alpha}$ controls the sampling noise in $\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$, with larger values producing more deterministic mixtures. Large values also downweight the prior component, ensuring the mixture closely approximates the pretrained model. The relative values of $\boldsymbol{\alpha}$ determine the expected mixture weights. To stabilise optimisation while preserving these relative weights, we clip the scale as follows:

$$\boldsymbol{\alpha} = \max\left(\epsilon, \frac{\boldsymbol{\alpha}}{\sum_i \alpha_i}\right) \times \min\left(\omega, \sum_i \alpha_i\right)$$

Here, ϵ prevents underflow and ω limits the total concentration to avoid overflow.

6.2.4 Fine-Tuning Loss

To fine-tune with NVIB regularisation, we augment the task-specific loss with KL divergence terms that penalise information flow through the latent representations. As in Chapter 3, this follows the variational information bottleneck (VIB) principle, which balances compression with task relevance. During training, latent representations are sampled from a nonparametric mixture defined by the NVIB layer, and the task loss $\mathcal{L}_T$ is computed using these samples.

To encourage generalisation, we add KL divergence terms between the approximate posterior and the prior over both the Gaussian components and Dirichlet weights. The total loss comprises three terms: the task loss $\mathcal{L}_T$, the Gaussian KL divergence $\mathcal{L}_G$, and the Dirichlet KL divergence $\mathcal{L}_D$, weighted by hyperparameters λ_G and λ_D, respectively.

Unlike the VAE settings in Chapters 3 and 4, where $\mathcal{L}_T$ was a reconstruction loss, here it corresponds to a supervised objective such as cross-entropy or mean squared error. We do not use the layerwise abstraction loss from Section 4.2.2, and instead average the regularisation over layers. The full fine-tuning objective is:

$$\mathcal{L} = \mathcal{L}_T \; + \; \lambda_D \mathcal{L}_D \; + \; \lambda_G \mathcal{L}_G$$

We refer to Section 3.3.1 for definitions of $\mathcal{L}_G$ and $\mathcal{L}_D$. This objective enables NVIB to regularise attention-based representations during fine-tuning, supporting both task performance and out-of-distribution generalisation.

6.3 Evaluation Across Modalities

To evaluate NVIB regularisation, we conduct controlled fine-tuning experiments across four modalities: speech, text, graphs, and vision, as illustrated in Figure 6.2. We compare to models that are first pretrained and then fine-tuned using empirical risk minimisation (ERM) with task-specific loss functions. For readability, details on datasets, model architectures, and evaluation metrics are grouped and presented in the respective experiment sections.

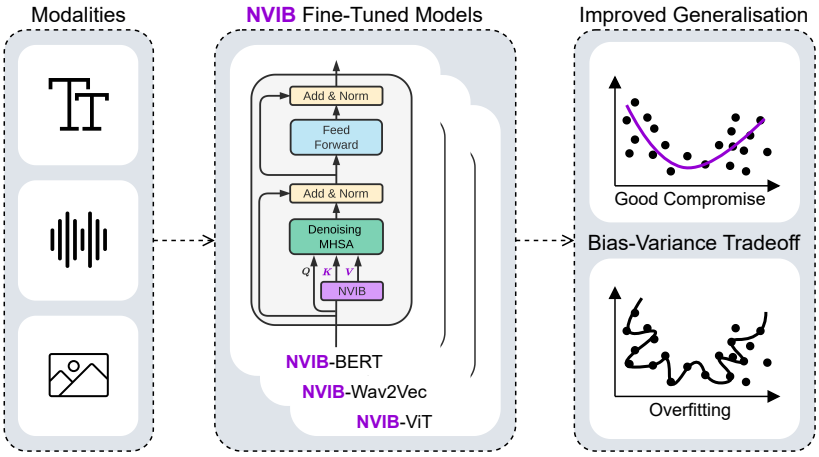


Figure 6.2: NVIB improves generalisation across diverse modalities. **Left:** Input data from four independent modalities: text, graphs (via text), speech, and vision. **center:** Pretrained models (BERT, Wav2Vec, ViT) are reinterpreted with NVIB layers and fine-tuned with regularisation. **Right:** NVIB reduces overfitting and improves generalisation by controlling the bias–variance trade-off.

Baselines For simplicity and to maintain uniformity across experiments, we define a set of fine-tuned baselines, avoiding modality-specific alternatives. These baselines include models trained without regularisation and models with dropout regularisation. To ensure consistency with standard practices, we use the predefined dropout rate of 0.1 for all pretrained models. Dropout is a suitable baseline for NVIB regularisation, as it is widely used and effective, seamlessly integrates into pretrained models, and introduces noise into both embeddings and attention mechanisms. All experiments are conducted on a consumer-grade NVIDIA RTX 3090 (24GB) GPU, with smaller Transformer models chosen to reduce computational costs.

Initialisation of NVIB layers The initialisation ensures empirical equivalence with each pretrained model after adjusting $(\tau_\sigma^2, \tau_\alpha)$, allowing the attention weights to ignore the prior component in NVIB layers. While in Chapter 5 Fehr and Henderson (2024) empirically initialise the prior component, we simplify this by setting $\boldsymbol{\mu}^p = \mathbf{0}$, $(\boldsymbol{\sigma}^p)^2 = \mathbf{1}$, and $\alpha_0^p = 1$. During fine-tuning, $\boldsymbol{\mu}^p$ remains learnable, while $(\boldsymbol{\sigma}^p)^2$ and α_0^p are fixed. However, stacking NVIB across layers in deeper models reduces equivalence precision. In such cases, NVIB is omitted from the later layers of the model. The parameters τ_σ and τ_α influence equivalence during training and evaluation.τ_σ controls initial Gaussian noise during fine-tuning but is unnecessary for evaluation equivalence, as mean embeddings are used. τ_α reweights Dirichlet pseudo-counts to ensure input embeddings outweigh the prior in attention.

Fine-tuning hyperparameters Following from Chapters 3 (Henderson and Fehr, 2023) and 4 (Behjati et al., 2023), we use a single sample per mixture component but omit the conditional prior, which is used during training from scratch to prevent posterior collapse. The KL divergence terms are weighted by λ_G and λ_D, respectively. For each experiment, we tune these hyperparameters using a log-scaled grid search over the values 10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, and 10^{-7}. We tie λ_G and λ_D, selecting the optimal value based on validation performance. Further hyperparameter details are provided in Appendix D.1.

6.3.1 Speech Out-of-Distribution Evaluation

Language identification and automated assessment of speech are crucial tasks in the development of audio transmission systems, but are challenging due to many factors related to: the acoustic environment; variation in recording hardware and software; speaker characteristics; and evaluation conditions (Gierlich and Kettler, 2006; Chinen, 2021; Cooper et al., 2022). The prediction of perceived speech quality is formulated as a regression task to estimate the scores of human listeners (ITU-T, 1996), whereas language identification is a classification task given an audio sample. Given the diverse array of factors that can impact speech, generalisation is essential in these tasks.

Speech quality assessment. We fine-tune and evaluate on the NISQA (Mittag et al., 2021) dataset, which contains English speech recordings from real-world scenarios, including live calls with network degradations and simulated audio distortions. This presents a realistic and challenging benchmark for quality assessment. For out-of-distribution (OOD) evaluation, we use the TencentWithReverberation (Tencent) dataset (Yi et al., 2022), a Chinese speech corpus that introduces additional complexity through both simulated and real reverberation, as well as varying labelling conditions. These settings reflect practical challenges encountered in deployed voice systems. Following ITU-T (2020), we report Pearson's correlation coefficient (PCC) and root-mean-square error after polynomial mapping (RMSE MAP).

We fine-tuned the pretrained Wav2vec2-base model[I] (Baevski et al., 2020), a 12-layer Transformer encoder (95M parameters), using mean-squared-error (MSE) loss. Fine-tuning was conducted with the Adam optimizer (Kingma and Ba, 2014), a constant learning rate of $1e^{-5}$, a batch size of 16, and for 5 epochs. NVIB was applied to layers 0–10, with projections initialised using $\tau_\sigma = 0.1$ and $\tau_\alpha = 10$. The best-performing model used $\lambda_G = \lambda_D = 1e^{-2}$. Table 6.1 shows that NVIB regularisation achieves the highest correlation on the in-distribution (ID) data. On the OOD dataset, NVIB regularisation achieves comparable generalisation improvements while exhibiting a lower standard deviation.

Table 6.1: Speech quality assessment for NISQA (ID) and Tencent (OOD). Average test results (0–1) are reported with standard deviation across 5 seeds.

	NISQA (ID)		Tencent (OOD)	
Model	**PCC** (↑)	**RMSE MAP** (↓)	**PCC** (↑)	**RMSE MAP** (↓)
$W2V2_{Base}$	0.89 (0.02)	0.42 (0.03)	0.80 (0.01)	0.54 (0.01)
with Dropout	0.89 (0.01)	0.43 (0.01)	**0.83** (0.03)	**0.51** (0.04)
with NVIB	**0.90** (0.01)	**0.41** (0.02)	**0.83** (0.02)	**0.51** (0.03)

Speech language identification. We fine-tune our models on the CommonLanguage (Ravanelli et al., 2021) dataset, which includes 22K audio clips from 45 languages. We evaluate on two OOD datasets with overlapping language sets: FLEURS (Conneau et al., 2023), containing read speech in 27 languages, and VoxPopuli (Wang et al., 2021), which comprises 11 languages spoken in spontaneous settings. While FLEURS is acoustically and stylistically similar to CommonLanguage, VoxPopuli is a far more challenging. It features unscripted European Parliament recordings, introducing real-world complexity through multilingual variation, background noise, diverse regional accents, and short utterances with limited context. These properties make it a robust test for generalisation in speech-based language identification.

[I] https://huggingface.co/facebook/wav2vec2-base

We fine-tuned the pretrained Wav2vec2-large model[II] (Baevski et al., 2020), a 24-layer Transformer encoder (317M parameters), using cross-entropy loss for the language identification classification task. Fine-tuning was performed with the AdamW optimizer (Loshchilov and Hutter, 2019), a learning rate of $3e^{-5}$, a scheduler with linear warm-up and decay, a batch size of 4, and for 10 epochs with mixed precision (16-bit) and gradient norm clipping of 1. NVIB was applied to layers 0–16, with projections initialized using $\tau_\sigma = 0$ and $\tau_\alpha = 10$. The best-performing model used $\lambda_G = \lambda_D = 1e^{-7}$. Table 6.2 reports the F1 classification scores, showing that NVIB matches ID performance and outperforms the dropout-regularised baseline on the OOD datasets.

Table 6.2: Language identification for CommonLanguage (ID), FLEURS (OOD) and VoxPopuli (OOD) speech datasets. Average test F1 scores (0–1) are reported with standard deviation across 5 seeds.

Model	CommonLanguage (ID) **F1** (↑)	FLEURS (OOD) **F1** (↑)	VoxPopuli (OOD) **F1** (↑)
$\text{W2V2}_{\text{Large}}$	**0.82** (0.01)	0.90 (0.02)	**0.86** (0.02)
with Dropout	0.81 (0.01)	0.90 (0.01)	0.82 (0.02)
with NVIB	**0.82** (0.01)	**0.91** (0.02)	0.85 (0.02)

6.3.2 Text Out-of-Distribution Classification

We evaluate on the CivilComments (CC) dataset (Borkan et al., 2019), which is part of the WILDS benchmark (Koh et al., 2021), a curated collection of datasets explicitly designed to capture real-world distribution shifts. CC addresses the task of toxicity classification in online comments, which is a socially important application for monitoring harmful content on the internet. This binary classification task includes a structured subpopulation shift across eight demographic identity groups. While the training and test domains share label space and features, their group proportions differ, reflecting demographic imbalances that arise in deployment. Following prior work, we measure generalisation by the worst-group (WG) accuracy, which is the classification performance on the least well-performing subpopulation.

We fine-tuned the pretrained TinyBERT[III] model (Turc et al., 2019), a two-layer Transformer encoder (5M parameters), using cross-entropy loss. Fine-tuning was performed with the AdamW optimizer (Loshchilov and Hutter, 2019), a constant learning rate of $5e^{-5}$, a batch size of 1024, and for 50 epochs with mixed precision (16-bit) and gradient norm clipping of 0.1. NVIB was applied to all layers, with projections initialized using $\tau_\sigma = 0.1$ and $\tau_\alpha = 1$, and a linear KL annealing warmup was used during fine-tuning. The best-performing model used $\lambda_G = \lambda_D = 1e^{-1}$.

[II] https://huggingface.co/facebook/wav2vec2-large
[III] https://huggingface.co/google/bert_uncased_L-2_H-128_A-2

Table 6.3 shows the generalisation improvement of this task through regularisation. On average, NVIB regularisation improves OOD generalisation over the unregularised baseline, though it remains less effective than dropout.

Table 6.3: Text classification on CC train (ID) and test (OOD). Average accuracy (%) is reported across 5 seeds with standard deviation and the *best* OOD model.

	CC Train (ID)	CC Test (OOD)
Model	**WG** (↑)	**WG** (↑)
$\text{BERT}_{\text{Tiny}}$	78.12 (14.33) *99.00*	49.14 (5.56) *61.03*
with Dropout	**91.05** (1.49) *91.16*	**60.10** (3.11) *63.97*
with NVIB	80.12 (10.69) *76.30*	55.01 (6.15) *61.03*

However, introducing sparsity in the attention keys based on their attention magnitude, as shown in Figure 6.3, improves OOD accuracy and sustains it across a wide range of sparsity levels. NVIB naturally induces key sparsity by reducing the weight of embeddings relative to the prior component during attention calculations. To remove keys, we mask embeddings with the lowest average attention magnitudes.

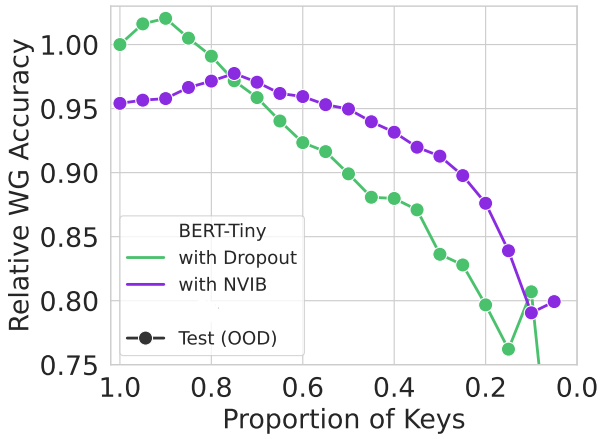


Figure 6.3: Worst-group (WG) test (OOD) accuracy as a function of attention key sparsity for the best OOD models, relative to dropout without sparsity.

Further inspection of the attention patterns in Figure 6.4 reveals a clear focus on toxic words as spurious keys are dropped and attention shifts to the prior token. The alignment with toxic content becomes more pronounced as sparsity increases. Additional examples can be found in Appendix Figures D.1, D.2 and D.3.

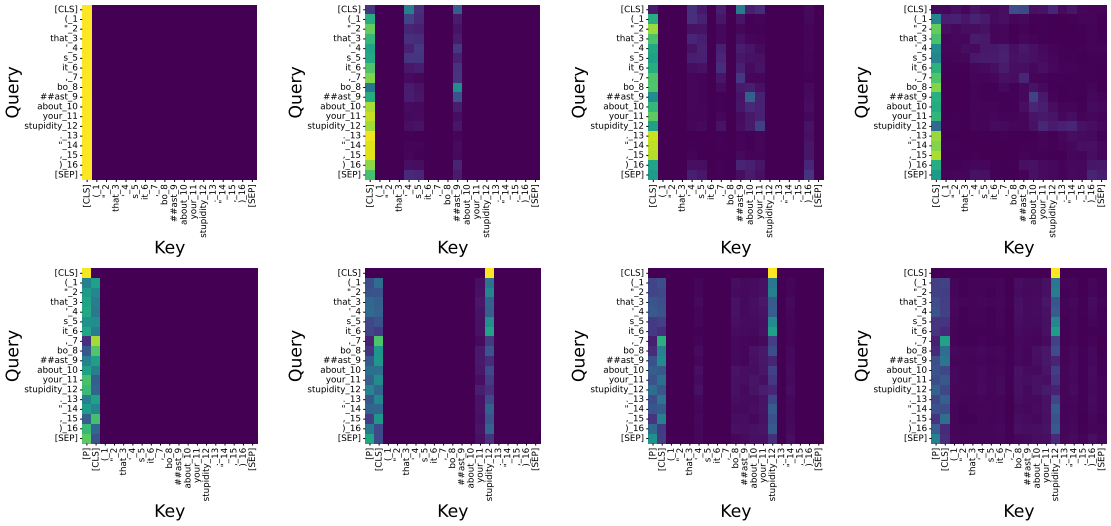


Figure 6.4: Attention plots from the best models on CivilComments, showing a single head of the last encoder layer. Left-Right: Proportion of keys retained [0.1, 0.25, 0.5, 1.0]. Top: with Dropout. Bottom: with NVIB. Sentence: ('that's it, boast about your stupidity.'). NVIB emphasizes 'boast' and 'stupidity'.

6.3.3 Graph Link Prediction

Link prediction is a graph-based problem that involves predicting whether a link exists between two nodes in a graph. This is widely used for recommendation and prediction in social networks, citation links and biological interactions (Kumar et al., 2020; Xia et al., 2021). We build upon the BERT for Link Prediction (BLP) model (Daza et al., 2020) which operates on a set of triples (h, r, t), where h and t represent the head and tail node, while r represents the relation between those two nodes.

We evaluate on the FB15k-237 dataset (Daza et al., 2020). This dataset follows an inductive setting, where new entities and triples are dynamically incorporated into the graph during evaluation. We evaluate the models by querying them with $(h, r, ?)$ and $(?, r, t)$ triples, and assess their performance using two metrics: Mean Reciprocal Rank (MRR), which measures the model's ability to rank the correct triple, and H@k, which calculates the proportion of correct triples ranked within the top-k results. We fine-tuned the pretrained TinyBERT[IV] model (Turc et al., 2019), a two-layer Transformer encoder (5M parameters), using a distance-based TransE loss function. Fine-tuning was performed with the RAdam optimizer (Liu et al., 2020b), a cosine learning rate scheduler with a value of $8e^{-5}$, a batch size of 256, and for 40 epochs with mixed precision (16-bit) and gradient norm clipping of 1. NVIB was applied to both layers, with projections initialised using $\tau_\sigma = 0.1$ and $\tau_\alpha = 1$. The best-performing model used $\lambda_G = \lambda_D = 1e^{-3}$.

[IV] https://huggingface.co/google/bert_uncased_L-2_H-128_A-2

Table 6.4 presents the test set results, which highlights the advantage of the NVIB-regularised model over typical regularisation methods like dropout. This advantage may stem from the presence of new entities in the head or tail positions, which require a higher level of generalisation.

Table 6.4: Graph link prediction on FB15k-237. Test set ranking metrics (0–1) are reported, based on the best model selected from validation set performance.

	FB15k-237			
Model	**MRR (↑)**	**H@1 (↑)**	**H@3 (↑)**	**H@10 (↑)**
BLP-BERT$_{\text{Tiny}}$	0.164	0.100	0.175	0.288
with Dropout	0.162	0.097	0.172	0.288
with NVIB	**0.167**	**0.103**	**0.180**	**0.294**

6.3.4 Image Few-Shot Classification

Few-shot classification aims to train models capable of classifying images with limited labelled examples per category. Meta-learning (Vinyals et al., 2016) achieves this by meta-training on several *episodes*, enabling generalisation to new tasks with previously unseen classes. To generalise effectively, the classifier must transfer knowledge from the training distribution to unseen testing distributions while avoiding spurious correlations and shortcuts (Zheng et al., 2024; Zhang et al., 2024).

The following experiments were conducted within a meta-learning-based few-shot classification framework (Hu et al., 2022b), using the pretrained DeiT-Small[V] (Touvron et al., 2021), a 12-layer Transformer encoder (22M parameters), with cross-entropy loss. Fine-tuning was performed using the AdamW optimizer (Loshchilov and Hutter, 2019), a constant weight decay of 0.05, and a linear warm-up with cosine decay learning rate scheduler $1e^{-4}$. The model was trained for 50 epochs with mixed precision (16-bit) and a batch size of 1. For classification, we used the prototypical network (ProtoNet) (Snell et al., 2017), which dynamically creates class centroids for each episode and performs nearest centroid classification (Hu et al., 2022b). In this experiment we initialised the prior $\mu^p = \mathbf{0}$ and did not allow it to be learnable.

Few-shot in-distribution. We evaluate the ID performance using the CIFAR-FS (Bertinetto et al., 2019) dataset. Following Hu et al. (2022b), we conduct experiments in a 5-way, 5-shot setting. Each episode consists of a "support set" with 5 classes and 5 samples per class for training, and a "query set" with 5 classes and 15 examples per class for testing. The experiment includes 2000 episodes for meta-training and 2000 episodes for testing. NVIB was applied to layers 0–5, with projections initialized using $\tau_\sigma = 0$ and $\tau_\alpha = 0$. The best-performing model used $\lambda_G = \lambda_D = 1e^{-2}$.

[V] https://huggingface.co/facebook/deit-small-patch16-224

Table 6.5 reports the average classification accuracy and standard deviation over all test episodes for CIFAR-FS in few-shot classification. Compared to the baseline and Dropout, we observe that NVIB regularisation improves accuracy with lower variance across all test episodes.

Table 6.5: Image classification on CIFAR-FS (ID). Test episodes accuracy (%) with standard deviation.

	CIFAR-FS (ID)	
Model	**Acc** (↑)	**Std** (↓)
$\text{DeiT}_{\text{Small}}$	93.57	5.71
with Dropout	93.55	5.61
with NVIB	**93.88**	**5.58**

Few-shot out-of-distribution. To evaluate the OOD few-shot classification performance, we use the Meta-Dataset (Triantafillou et al., 2019). This benchmark is a diverse set of 10 image datasets, including, ImageNet-1k, MSCOCO (COCO), Traffic Signs (Sign), Describable Textures (DTD), FGVCx Fungi (Fungi), Omniglot, VGG Flower (Flower), CUB-200-2011 (CUB), FGVCAircraft (Acraft) and QuickDraw (QDraw). We meta-train the models on ImageNet-1k and then meta-test them on the remaining datasets.

We follow the methodology outlined in Hu et al. (2022b), where the number of ways sampled ranges from 5 to 50, with a maximum support size of 500 and a maximum query size of 10. For our Transformer encoder, we apply NVIB to layers 0–5, with projections initialised using $\tau_\sigma = 0$ and $\tau_\alpha = -3$. The best-performing model used NVIB regularization parameters of $\lambda_G = \lambda_D = 1e^{-3}$. Figure 6.5 shows that the NVIB-regularised model achieves the highest performance on 6 out of 9 OOD datasets and outperforms the dropout-regularised model in 7 out of 9 cases.

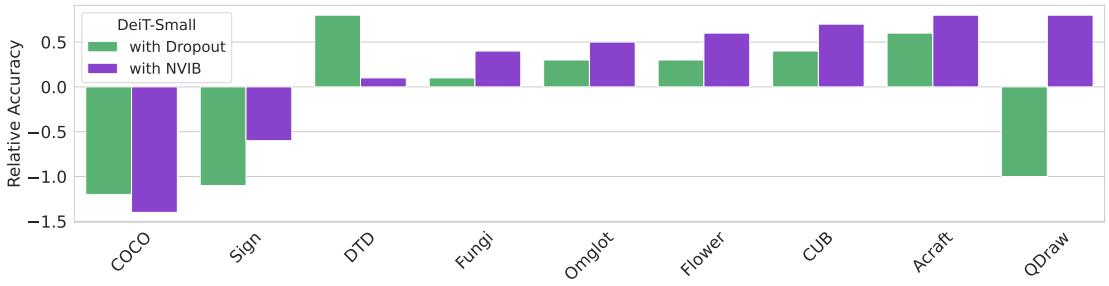


Figure 6.5: Percentage point improvement in test accuracy relative to the unregularised baseline on the Meta-Dataset benchmark (OOD).

6.3.5 Image Privacy Classification

Image privacy classification is a crucial task for preventing the leakage of sensitive visual content. Models must be both accurate and robust, generalising across imbalanced categories, ambiguous visual cues, and noisy inputs to reliably detect private information under real-world conditions. We evaluate on the PrivacyAlert dataset (Zhao et al., 2022), which contains user-generated images from Flickr labelled as either private or public. The dataset poses significant challenges for generalisation: only 25% of images are private, and these frequently depict cluttered scenes with multiple objects, occlusions, and diverse visual contexts. It also exhibits class imbalance and label noise, particularly in under-represented categories such as Medical and Personal Information.

We fine-tuned the pretrained DeiT-Tiny [VI] (Touvron et al., 2021), a 12-layer Transformer encoder (5M parameters), with cross-entropy loss. Fine-tuning was conducted with the AdamW optimizer (Loshchilov and Hutter, 2019), a constant learning rate of $5e^{-6}$, a batch size of 32, and for 80 epochs with mixed precision (16-bit). NVIB was applied to all layers, with projections initialized using $\tau_\sigma = 0.5$ and $\tau_\alpha = 8$. The best-performing model used $\lambda_G = \lambda_D = 1e^{-3}$. We evaluate classification performance using F1 scores. Table 6.6 shows that NVIB outperforms the dropout-regularised baseline.

Table 6.6: Image privacy classification on PrivacyAlert. Average test F1 scores (%) are reported with standard deviation across 5 seeds.

	PrivacyAlert (ID)	
Model	**F1** (↑)	**Std** (↓)
$\text{DeiT}_{\text{Tiny}}$	78.41	**0.10**
with Dropout	76.26	2.47
with NVIB	**79.40**	0.72

We assess the robustness by perturbing images with zero-mean Gaussian noise at varying standard deviations. The robustness is measured using the attack success rate (ASR), which is the proportion of correctly classified images that are misclassified after perturbation. Figure 6.6 indicates that NVIB achieves robustness comparable to dropout while offering improvements over an unregularised vision baseline.

[VI] https://huggingface.co/facebook/deit-tiny-patch16-224

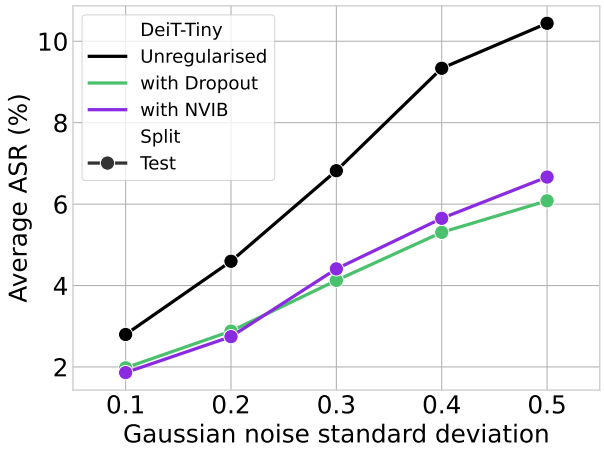


Figure 6.6: Average attack success rate (ASR) on privacy classifiers, reported across 5 models with varying Gaussian noise standard deviations. A lower ASR indicates greater model robustness.

6.4 Discussion

NVIB provides modest but consistent gains in generalisation across fine-tuning tasks. It outperforms unregularised models and often surpasses dropout, particularly in Graph Link Prediction (Table 6.4), Few-Shot Meta-Learning (Figure 6.5), and Image Privacy Classification (Table 6.6). NVIB also shows lower variance than dropout (Tables 6.1, 6.5, 6.6). We attribute these gains to NVIB's ability to distinguish signal from noise via structured noise injection into latent representations. This is especially beneficial in noisy or imbalanced settings such as speech quality and privacy tasks (Tables 6.1, 6.6). Its Bayesian formulation captures uncertainty during optimisation, supporting robustness.

NVIB further improves generalisation by reweighting attention through prior tokens, reducing reliance on spurious features. This effect is evident in tasks requiring extrapolation to unseen data (Table 6.4, Figure 6.5) and in attention maps that highlight informative tokens (Figures 6.4, D.1, D.2 & D.3). NVIB also induces sparsity (Figure 6.3) and maintains performance under perturbations (Figure 6.6).

6.5 Conclusion

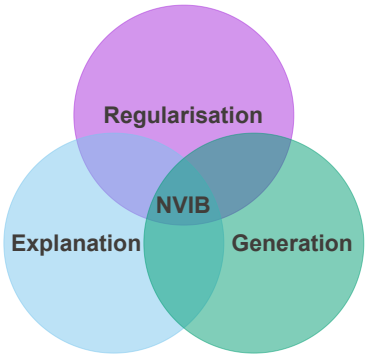
This chapter addressed how generalisable representations can be induced during fine-tuning by introducing Nonparametric Variational Information Bottleneck (NVIB) regularisation. We extended NVIB to fine-tuning pretrained models and demonstrated improved generalisation across a range of modalities and architectures. Key contributions include a learnable prior mean per layer for greater adaptability, Dirichlet parameter clipping for training stability, and a simplified denoising attention function for inference.

We find that NVIB consistently improves generalisation, particularly in noisy or imbalanced settings, while maintaining low variance across runs. Though performance gains are modest, they are reliable and often surpass dropout. NVIB acts as a structured regulariser, encouraging sparsity and suppressing spurious correlations in the learned representations. Attention visualisations and sparsity patterns suggest that NVIB promotes more abstract and task-relevant token interactions.

We evaluate NVIB across models ranging from 5M to 317M parameters and observe no degradation due to scale. All experiments were run on a single RTX 3090 (24GB), which limited model scale. Extending NVIB to large language models remains a promising but non-trivial challenge, both computationally and architecturally, due to factors such as causal masking and relative positional encodings.

In summary, NVIB provides a general, probabilistically grounded approach to regularising attention-based models during fine-tuning. It induces generalisable representations by injecting structured, information-theoretic noise into embeddings and attention. This enables the model to capture meaningful signal and reduce reliance on spurious features arising from noise. The result is improved robustness and interpretability across diverse tasks and modalities. In the final chapter, we reflect on the broader contributions of this thesis and outline promising directions for advancing NVIB further.

7 Conclusions & Going Beyond with NVIB



This thesis presents a variational Bayesian perspective on attention, showing that Transformer representations can be regularised, explained, and generalised through latent variable inference. Attention is not merely architectural, but inferential: it can be modelled as posterior inference over nonparametric latent variables. This view clarifies why attention generalises well—it performs denoising by abstracting signal from noise in unbounded, permutation-invariant sets. NVIB is the tool that facilitates this perspective, enabling a unified approach across pretraining, reinterpretation of pretrained models, and fine-tuning across diverse modalities. It supports three core facets of representation learning: **regularisation**, **explanation**, and **generation**. The resulting representations are abstract, generalisable, sparse, interpretable, and robust. By treating attention as Bayesian posterior inference, we gain tools to shape and understand the latent structure it encodes. This thesis is not a conclusion, but a foundation: latent variable models of attention offer new paths for advancing representation learning.

Chapter Contributions Each chapter addressed a core research question, advancing our understanding of how latent variables can structure and improve attention-based models. In Chapter 3, we introduced NVIB, a nonparametric variational latent variable framework that formulates attention as inference over mixture distributions. This formulation captures two core properties of attention: permutation invariance and variable input size. It enables a Transformer-based VAE that learns sparse and regularised representations and supports generation through a smooth, structured latent space. In Chapter 4, we extended NVIB across stacked self-attention layers to induce abstraction and interpretability in Transformers. The model learned to allocate fewer vectors at deeper layers, forming a hierarchy of representations without supervision. These abstractions aligned with linguistic structure and produced sparse, robust, and interpretable attention patterns. In Chapter 5, we extended NVIB to multi-head attention and applied it to pretrained Transformers. This enabled a post-training reinterpretation of attention as variational inference, using identity initialisation and empirical priors to control uncertainty. We found that pretrained Transformers encode information in a way that is well modelled by nonparametric variational distributions. Without additional training, NVIB regularised

attention by down-weighting unreliable information and improved out-of-distribution generalisation on translation and summarisation tasks. In Chapter 6, we extended NVIB to fine-tuning across diverse pretrained models and modalities. We introduced architectural improvements, which enable NVIB to act as a structured regulariser during fine-tuning. The method induced sparse, robust representations and improved out-of-distribution generalisation in speech, text, graphs, and vision. This shows that NVIB can reliably induce generalisable representations across tasks and domains.

Limitations While NVIB provides a general and flexible framework, this thesis focused on moderate-scale models, limited training regimes, and a restricted scope of tasks. In Chapter 3, we used small datasets and short sequences to develop a proof-of-concept Transformer-VAE in the training-from-scratch setting. In Chapter 4, we pretrained a Transformer encoder at small scale and evaluated only on English text with character-level tokenisation. Chapter 5 scaled to be able to reinterpret larger pretrained models but was limited to encoder-decoder architectures and post-training regularisation for summarisation and translation. In Chapter 6, we extended NVIB to the fine-tuning paradigm but prioritised breadth of modalities over depth and scale. Across all chapters, computational constraints limited both model size and experimental scope. Integrating NVIB with decoder-only LLMs and hardware-aware attention mechanisms to harness the sparisty remains an open challenge, for both pretraining and fine-tuning.

Going Beyond The limitations discussed across the chapters point to several promising directions for future research, structured around the three core facets of NVIB: **regularisation**, **explanation**, and **generation**. To address the scale and sequence length constraints of Chapter 3, future work can extend NVIB to decoder-only language models, long-context settings, and retrieval-augmented generation. By sparsifying attention and suppressing uninformative content, NVIB may improve performance on needle-in-a-haystack problems—where relevant information must be retrieved from long contexts; especially when paired with hardware-efficient mechanisms such as flash attention or mixture-of-experts routing. To go beyond the limited scope of Chapter 4, future work can explore how NVIB-induced abstractions support interpretability beyond the character level, including grammar, syntax, opinion summarisation across entities, and higher-level reasoning tasks such as argument structure extraction. These abstractions may be especially valuable in low-resource and out-of-distribution settings, where generalisation depends on uncovering low-dimensional latent structure. Building on the smooth latent space observed in Chapters 3 and 5, NVIB also offers a generative formulation over unbounded latent sets. This supports latent-space sampling and denoising, bridging attention mechanisms with diffusion-style inference over sets. As scaling shows diminishing returns in today's LLMs, we face a less bitter lesson: inductive biases, such as regularisation, are becoming increasingly important. NVIB provides a principled, theory-driven explanation of model behaviour and bridges to diffusion-style inference, supporting the development of more controllable and generalisable generative models for the future of AI.

A Appendix for Chapter 3

A.1 Hyperparameter Tuning

The models are trained on the Wikitext-2 training dataset using the loss from equation 3.7. They are tuned on the validation dataset with the aim to be able to both reconstruct and generate output. All combinations of the following hyperparameters were considered in a grid search for the respective models:

- $\lambda'_G = \{1,\ 1e^{-1},\ 1e^{-2},\ 1e^{-3},\ 1e^{-4},\ 1e^{-5},\ 0\}$
- $\lambda'_D = \{10,\ 1,\ 1e^{-1},\ 1e^{-2},\ 1e^{-3},\ 1e^{-4},\ 1e^{-5},\ 0\}$
- $\alpha^{\Delta} = \{1,\ 0.75,\ 0.5,\ 0.4,\ 0.3,\ 0.2,\ 0.1,\ 0\}$
- $\kappa^{\Delta} = \{1,\ 2,\ 5\}$
- $S = \{0.9,\ 0.8,\ 0.75,\ 0.5,\ 0.25\}$
- $P = \{\text{mean},\ \text{max},\ \text{one}\}$

where λ'_G and λ'_D are the weights on the Gaussian and Dirichlet KL divergences for all variational models, respectively. The α^{Δ} and κ^{Δ} are NVAE specific parameters and represent the conditional prior parameter and number of samples per component. The stride parameter S for the VTS model results in $1 - S$ proportion of vectors being kept. Finally, P is the pooling method for the single vector model VTP.

Baselines Empirically we found the best KL divergence parameter for VT, VTP and VTS is $\lambda'_G = 1e^{-2}$ and using max pooling for VTP. All stride parameters are considered to adjust the number of vectors. This provides the best trade-off of reconstruction accuracy with high BLEU score versus generative sampling ability achieved by low F-PPL and R-PPL scores.

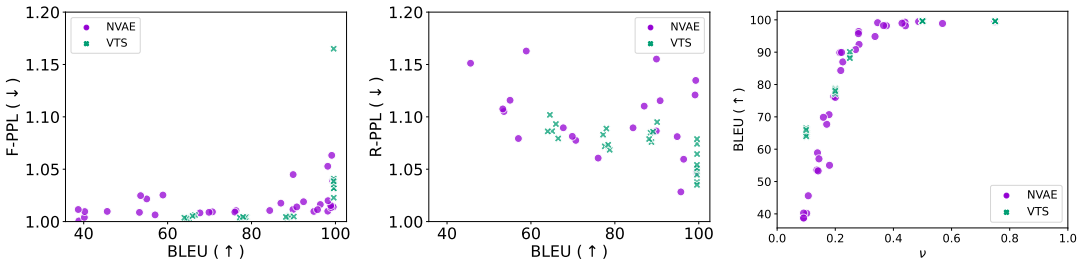
NVAE The hyperparameter tuning for NVAE aims to discover models which: neither collapse to a single vector nor use all vectors, reconstruct accurately, and are able to sample effectively from the prior by achieving low F-PPL and R-PPL scores. Empirically we find the parameters $\lambda'_G = 1e^{-3}$, $\lambda'_D = 1$ and $\kappa^{\Delta} = 1$ to produce the best trade off between reconstruction accuracy and generative ability. The α^{Δ} parameter is able to control the proportion of vectors (See Figure 3.10) retained and $\kappa^{\Delta} = 1$ provides an efficient sampling of the model (See Figure 3.9).

Validation results Table A.1 displays the validation reconstruction and generation results across 5 seeds for the best performing parameters. The VT model is able reconstruct well, but a high F-PPL score suggest a poor fluency of generated text and large variation. The VTP models show that a single-vector bottleneck is insufficient to reconstruct. Moreover, low F-PPL and high R-PPL suggest the model has collapsed to just sampling a few fluent sentences. The VTS models show that some fixed proportions ν of vectors retained result in good overall performance. NVAE is able to find models with comparable average proportion ν of vectors retained to those hand-coded in the VTS models. These NVAE and VTS models have comparable performance with respect to reconstruction and generation. However, the NVAE models have notably more variance of metrics across seeds.

			Reconstruction			**Generation**	
Model		ν	**BLEU** (↑)	**PPL** (↓)		**F-PPL** (↓)	**R-PPL** (↓)
VT		1.00	99.63±0.00	1.00	±0.00	1.96 ±0.89	1.06 ±0.02
VTS	$S = 0.5$	0.50	99.59 ±0.01	1.00	±0.00	1.03 ±0.01	1.06 ±0.01
VTS	$S = 0.75$	0.25	88.92 ±0.73	3.74	±0.37	1.00 ±0.00	1.08 ±0.01
VTS	$S = 0.8$	0.20	77.99 ±0.63	18.78	±1.41	1.00 ±0.00	1.08 ±0.01
VTS	$S = 0.9$	0.10	65.20 ±1.02	66.71	±11.46	1.00 ±0.00	1.09 ±0.01
VTP	$P = max$	0.05*	46.36 ±0.40	1659.24	±70.28	1.01 ±0.00	1.21 ±0.03
VTP	$P = mean$	0.05*	42.94 ±0.49	2425.25	±97.65	1.09 ±0.03	1.35 ±0.02
VTP	$P = one$	0.05*	38.33 ±1.08	2902.39	±456.97	1.00 ±0.00	1.29 ±0.03
NVAE	$\alpha^{\Delta} = 1$	0.50 ±0.15	98.90 ±0.67	1.07	±0.07	1.03 ±0.02	1.50 ±0.65
NVAE	$\alpha^{\Delta} = 0.75$	0.40 ±0.10	98.19 ±1.87	1.15	±0.21	1.02 ±0.02	1.15 ±0.10
NVAE	$\alpha^{\Delta} = 0.5$	0.27 ±0.06	92.78 ±4.61	2.14	±1.06	1.02 ±0.00	1.15 ±0.09
NVAE	$\alpha^{\Delta} = 0.4$	0.23 ±0.06	83.93 ±14.58	29.75	±60.42	1.02 ±0.01	1.11 ±0.05
NVAE	$\alpha^{\Delta} = 0.3$	0.18 ±0.02	67.01 ±-10.30	119.97	±151.43	1.01 ±0.01	1.12 ±0.09
NVAE	$\alpha^{\Delta} = 0.2$	0.13 ±0.03	54.02 ±14.67	1145.56	±1490.82	1.01 ±0.01	1.26 ±0.27
NVAE	$\alpha^{\Delta} = 0.1$	0.11 ±0.02	44.46 ±6.59	1635.51	±1122.01	1.01 ±0.00	1.21 ±0.10

Table A.1: Results for regularisation and generation on validation Wikitext-2 averaged over 5 seeds. The average proportion of latent vectors retained during evaluation is reported by ν. *The VTP models only use a single vector.

Figures A.1a, A.1b and A.1c visually display the reconstruction versus generation trade-off for the validation data across seeds. The best models have both low generation perplexities and high BLEU scores whilst dropping a large proportion of vectors. Figure A.1c shows that the NVAE model is able to dynamically reduce the number of vectors and still reconstruct, comparably to the VTS models. We notice that their exist some NVAE models that have a good reconstruction-generation trade-off but are less clustered than the VTS models. Note that the R-PPL and F-PPL plot limits are cropped at 1.2 to focus on the higher performing models.



(a) Validation BLEU vs F-PPL. (b) Validation BLEU vs R-PPL. (c) Validation BLEU vs proportion of retained vectors ν.

Figure A.1: Reconstruction versus generation trade-off and regularisation for Wikitext-2 validation.

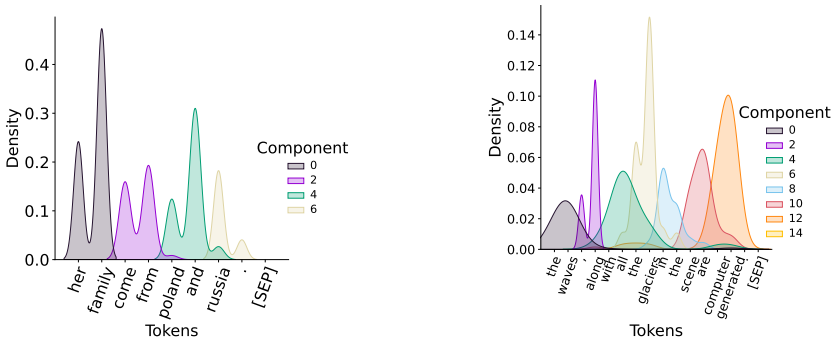
Test results The best seed models from Table A.1 are selected for the baselines and NVAE and then evaluated on the test set and shown in Table A.2 and a subset of this $\alpha^{\Delta} = \{0.75,\ 0.4,\ 0.3,\ 0.2\}$ is plotted in Figure 3.8.

			Reconstruction		**Generation**	
Model		ν	**BLEU** (↑)	**PPL** (↓)	**F-PPL** (↓)	**R-PPL** (↓)
VT		1.00	99.63	1.00	1.37	1.11
VTS	$S = 0.5$	0.50	99.61	1.00	1.03	1.05
VTS	$S = 0.75$	0.25	89.18	3.72	1.00	1.07
VTS	$S = 0.8$	0.20	79.51	15.51	1.00	1.06
VTS	$S = 0.9$	0.10	67.04	51.47	1.01	1.08
VTP	$P = max$	0.05*	48.940	1386.34	1.01	1.17
NVAE	$\alpha^{\Delta} = 1$	0.44	99.27	1.04	1.04	1.12
NVAE	$\alpha^{\Delta} = 0.75$	0.34	99.15	1.04	1.01	1.04
NVAE	$\alpha^{\Delta} = 0.5$	0.28	96.35	1.33	1.02	1.05
NVAE	$\alpha^{\Delta} = 0.4$	0.28	95.96	1.41	1.01	1.03
NVAE	$\alpha^{\Delta} = 0.3$	0.19	75.83	17.18	1.01	1.05
NVAE	$\alpha^{\Delta} = 0.2$	0.16	70.60	23.46	1.01	1.08
NVAE	$\alpha^{\Delta} = 0.1$	0.14	54.35	267.59	1.01	1.11

Table A.2: Results for regularisation and generation on test Wikitext-2. The average proportion of latent vectors retained during evaluation is reported by ν. *The VTP models only use a single vector.

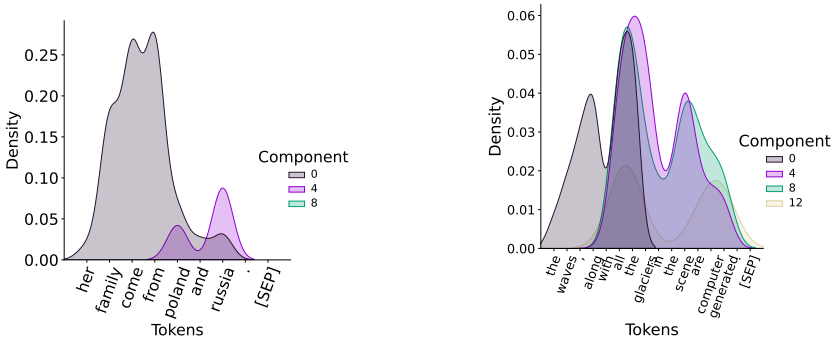
A.2 Alignment Analysis

This experiment highlights the problem of latent alignment in non-NVIB models, and evaluates alignment based on position, which we use in the interpolation experiments. We consider the VTS models from Table A.2 and use them to encode a sentence into their latent space. For each latent component retained by the VTS model, we perturb it with Gaussian noise and consider the resulting autoregressively decoded output. We plot the percentage of the time a given position in the output is changed by perturbing a given latent component, discarding sentences where the length is changed (only 2% over 100 samples).



(a) VTS $S = 0.5$ (b) VTS $S = 0.5$

Figure A.2: Latent vector alignment with generated output



(a) VTS $S = 0.75$ (b) VTS $S = 0.75$

Figure A.3: Latent vector alignment with generated output.

Figure A.2 shows that the latent space for the VTS $S = 0.5$ model are approximately aligned by location. However in Figure A.3, for more condensed representations where $S = 0.75$, there is an unclear alignment of latent vectors to their position. As discussed in Section 3.5.3, the use of mixture distributions instead of sets of vectors allows NVIB representations to avoid this problem of alignment.

A.3 Deriving the Factorised Dirichlet Process

In this section we derive an alternative factorisation of a DP which helps with the sampling method in Section 3.3.2.

Theorem A.3.1. *Let $F \sim \mathrm{DP}(G_0^q, \alpha_0^q)$ be a Dirichlet Process, represented via an infinite symmetric Dirichlet distribution:*

$$F = \sum_{k=1}^{\infty} \pi_k \, \delta_{\boldsymbol{z}_k},$$

$$\boldsymbol{\pi} \sim \lim_{\kappa_0 \to \infty} \mathrm{Dir}\left(\tfrac{\alpha_0^q}{\kappa_0}, \ \ldots, \ \tfrac{\alpha_0^q}{\kappa_0}\right),$$

$$\boldsymbol{z}_k \sim G_0^q \quad \text{for } k = 1, \ldots, \infty$$

Let $\boldsymbol{G}^q = (G_1^q, \ldots, G_c^q)$ be a finite partition of base distributions and $\boldsymbol{\alpha}^q = (\alpha_1^q, \ldots, \alpha_c^q)$ their associated concentration parameters, where c denotes the number of components, including the prior. Then a Dirichlet Process can be defined via its finite marginals over partitions Teh (2010):

$$\alpha_0^q = \sum_{i=1}^{c} \alpha_i^q, \qquad G_0^q = \sum_{i=1}^{c} \frac{\alpha_i^q}{\alpha_0^q} G_i^q.$$

*We can then define the equivalent **Factorised Dirichlet Process** $F_{\text{fact}} \sim \mathrm{FDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q)$ by:*

$$\boldsymbol{\rho} \sim \mathrm{Dir}(\alpha_1^q, \ldots, \alpha_c^q),$$

$$\tilde{F}_i \sim \mathrm{DP}(G_i^q, \alpha_i^q), \quad \text{for } i = 1, \ldots, c,$$

$$F_{\text{fact}} = \sum_{i=1}^{c} \rho_i \tilde{F}_i.$$

Then the two constructions are equivalent in distribution:

$$\mathrm{FDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q) = DP(G_0^q, \alpha_0^q).$$

Proof. Let $F \sim \mathrm{DP}(G_0^q, \alpha_0^q)$ be a Dirichlet Process, represented via an infinite symmetric Dirichlet distribution:

$$F = \sum_{k=1}^{\infty} \pi_k \, \delta_{\boldsymbol{z}_k},$$

$$\boldsymbol{\pi} \sim \lim_{\kappa_0 \to \infty} \mathrm{Dir}\left(\tfrac{\alpha_0^q}{\kappa_0}, \ \ldots, \ \tfrac{\alpha_0^q}{\kappa_0}\right),$$

$$\boldsymbol{z}_k \sim G_0^q \quad \text{for } k = 1, \ \ldots, \ \infty$$

where the weights $\boldsymbol{\pi}$ and vectors $\boldsymbol{z}_k$ are sampled independently.

For the vectors, we know that after generating an infinite number of $\boldsymbol{z}_k$ from G_0^q, a proportion of exactly $\frac{\alpha_i^q}{\alpha_0^q}$ will be generated from component G_i^q. For a finite number κ_0 of vectors, let κ_i be the number of samples $\boldsymbol{z}_k$ drawn from base distribution component G_i^q, for each $i = 1, \ldots, c$. Then:

$$\lim_{\kappa_0 \to \infty} \frac{\kappa_i}{\kappa_0} = \frac{\alpha_i^q}{\alpha_0^q}.$$

By the *exchangeability* property of Dirichlet Processes, they are invariant to the order of data points. Thus, we may relabel the κ_0 categories of $\text{Dir}\left(\frac{\alpha_0^q}{\kappa_0}, \ldots, \frac{\alpha_0^q}{\kappa_0}\right)$ so that the weights $\boldsymbol{\pi}$ can be partitioned as:

$$\boldsymbol{\pi} = (\pi_{11}, \ldots, \pi_{1\kappa_1}, \ \ldots \ , \pi_{c1}, \ldots, \pi_{c\kappa_c}),$$

where π_{ij} is the weight associated with the j-th vector from base distribution component i. Then the corresponding vectors are sampled as:

$$\boldsymbol{z}_{ij} \sim G_i^q \quad \text{for } i = 1, \ldots, c; \quad j = 1, \ldots, \kappa_i.$$

For finite κ_0, before taking the limit as $\kappa_0 \to \infty$, we define the total weight assigned to each partition as:

$$\rho_i = \sum_{j=1}^{\kappa_i} \pi_{ij}, \quad \text{for } i = 1, \ldots, c.$$

Letting $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_c)$, this forms the vector of total weights across the c components. By the *aggregation property* of Dirichlet distributions (Frigyik et al., 2010, Section 1.3) (See Section 2.3.3), we have:

$$\boldsymbol{\rho} \sim \text{Dir}(\alpha_1^q, \ldots, \alpha_c^q).$$

By the *neutrality* of the Dirichlet distribution (Frigyik et al., 2010, Section 2.2.2) (See Section 2.3.3), the total weights $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_c)$ and the normalised intra-partition weights,

$$\left(\frac{\pi_{i1}}{\rho_i}, \ldots, \frac{\pi_{i\kappa_i}}{\rho_i}\right) \quad \text{for } i = 1, \ \ldots, \ c$$

are independent. This implies that, conditioned on ρ_i, the normalized weights within partition i depend only on the relative values of π_{ij}. To derive their distribution, marginalise out all categories outside partition i by grouping them into a single category of weight

$1-\rho_i$, yielding:

$$(\pi_{i1},\ldots,\pi_{i\kappa_i},1-\rho_i) \sim \mathrm{Dir}\left(\tfrac{\alpha_0^q}{\kappa_0},\ldots,\tfrac{\alpha_0^q}{\kappa_0},\, \alpha_0^q\left(1-\tfrac{\kappa_i}{\kappa_0}\right)\right).$$

Following a formulation analogous to stick-breaking, we marginalise out the weight of the remaining category by integrating over ρ_i, yielding:

$$\left(\frac{\pi_{i1}}{\rho_i},\ldots,\frac{\pi_{i\kappa_i}}{\rho_i}\right) \sim \mathrm{Dir}\left(\tfrac{\alpha_0^q}{\kappa_0},\ldots,\tfrac{\alpha_0^q}{\kappa_0}\right).$$

Having derived the individual distributions, we now express the joint distribution for finite κ_0 in factorised form:

$$\begin{aligned}
\pi_{ij} &= \rho_i\,\pi'_{ij} && \text{for } i=1,\ldots,c;\ j=1,\ldots,\kappa_i,\\
\boldsymbol{\rho} &\sim \mathrm{Dir}(\alpha_1^q,\ldots,\alpha_c^q), &&\\
\boldsymbol{\pi}'_i &\sim \mathrm{Dir}\left(\tfrac{\alpha_0^q}{\kappa_0},\ldots,\tfrac{\alpha_0^q}{\kappa_0}\right) && \text{for } i=1,\ldots,c.
\end{aligned}$$

As $\kappa_0\to\infty$, we note that $\frac{\alpha_0^q}{\kappa_0}\to\frac{\alpha_i^q}{\kappa_i}$ for each i, giving the factorised form:

$$\begin{aligned}
\pi_{ij} &= \rho_i\,\pi'_{ij} && \text{for } i=1,\ldots,c;\ j=1,\ldots,\infty,\\
\boldsymbol{\rho} &\sim \mathrm{Dir}(\alpha_1^q,\ldots,\alpha_c^q), &&\\
\boldsymbol{\pi}'_i &\sim \lim_{\kappa_i\to\infty}\mathrm{Dir}\left(\tfrac{\alpha_i^q}{\kappa_i},\ldots,\tfrac{\alpha_i^q}{\kappa_i}\right) && \text{for } i=1,\ldots,c.
\end{aligned}$$

Thus, the DP weights can be equivalently expressed as those of a factorised Dirichlet process (FDP) constructed via the above decomposition.

Combining the weights and vectors, we obtain the distribution F_i over the weighted atoms in each partition i:

$$\begin{aligned}
\boldsymbol{z}_{ij} &\sim G_i^q && \text{for } i=1,\ldots,c;\ j=1,\ldots,\kappa_i,\\
\boldsymbol{\pi}'_i &\sim \lim_{\kappa_i\to\infty}\mathrm{Dir}\left(\tfrac{\alpha_i^q}{\kappa_i},\ldots,\tfrac{\alpha_i^q}{\kappa_i}\right) && \text{for } i=1,\ldots,c,\\
\tilde{F}_i &\sim \mathrm{DP}(G_i^q,\alpha_i^q) && \text{for } i=1,\ldots,c.
\end{aligned}$$

This completes the proof that, if $F\sim\mathrm{DP}(G_0^q,\alpha_0^q)$, then it can be expressed as a **Factorised**

Dirichlet Process $F_{\text{fact}} \sim \text{FDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q)$:

$$
\begin{aligned}
F_{\text{fact}} &= \sum_{i=1}^{c} \rho_i \, \tilde{F}_i, \\
\boldsymbol{\rho} &\sim \text{Dir}(\alpha_1^q, \ldots, \alpha_c^q), \\
\tilde{F}_i &\sim \text{DP}(G_i^q, \alpha_i^q) \qquad\qquad \text{for } i = 1, \ldots, c.
\end{aligned}
$$

Hence, we conclude that:

$$\text{FDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q) = \text{DP}(G_0^q, \alpha_0^q).$$

□

A.4 Deriving the Kullback–Leibler Divergence

In this section, we derive the KL divergence between the variational posterior and the prior. We also show that this divergence grows approximately linearly with the number of sampled vectors per component.

To compare the two distributions directly, we first express the prior in the same Bounded Factorised Dirichlet Process (BFDP) form as the posterior. This is possible without altering the distribution represented by the prior. Specifically, we replicate the prior base distribution G_0^p across $n+1$ components, assigning weights to match the structure of the posterior base mixture. This yields a prior in the form:

$$F_{\text{fact}}^p \sim \text{BFDP}(\boldsymbol{G}^p, \boldsymbol{\alpha}^{p'}, \kappa_0)$$

where, $\boldsymbol{G}^p = (G_0^p, \ldots, G_0^p)$ is $n+1$ identical base distributions, $\boldsymbol{\alpha}^{p'} = \left(\alpha_0^{p'} \frac{\alpha_1^q}{\alpha_0^q}, \ldots, \alpha_0^{p'} \frac{\alpha_{n+1}^q}{\alpha_0^q}\right)$, $\alpha_0^{p'}$ is the total concentration parameter, and κ_0 bounds the approximation by a function of the input. Expressing both prior and posterior as BFDPs with matched structure simplifies the KL divergence calculation, since the divergence decomposes into a sum over the KL divergences between corresponding factor pairs.

Theorem A.4.1. *Let the posterior be given by a bounded Factorised Dirichlet Process* $q(F \mid x) = \mathrm{BFDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q, \kappa_0)$*, and the prior rewritten in the same factorised form:*$p(F) = \mathrm{BFDP}(\boldsymbol{G}^p, \boldsymbol{\alpha}^{p'}, \kappa_0)$*, Then the KL divergence between the two can be approximated as:*

$$\mathrm{D}_{\mathbb{KL}}(q(F \mid x) \,\|\, p(F)) \approx \mathcal{L}_D + \mathcal{L}_G,$$

where:

$$\begin{aligned}\mathcal{L}_D &= \log \Gamma(\alpha_0^q) - \log \Gamma(\alpha_0^{p'}) \\ &\quad + (\alpha_0^q - \alpha_0^{p'}) \left(\psi\left(\tfrac{\alpha_0^q}{\kappa_0}\right) - \psi(\alpha_0^q)\right) \\ &\quad + \kappa_0 \left(\log \Gamma\left(\tfrac{\alpha_0^{p'}}{\kappa_0}\right) - \log \Gamma\left(\tfrac{\alpha_0^q}{\kappa_0}\right)\right),\end{aligned}$$

$$\mathcal{L}_G = \tfrac{1}{2}\kappa_0 \sum_{i=1}^{n+1} \tfrac{\alpha_i^q}{\alpha_0^q} \sum_{h=1}^{d} \left(\tfrac{(\mu_{ih}^q - \mu_h^p)^2}{(\sigma_h^p)^2} + \tfrac{(\sigma_{ih}^q)^2}{(\sigma_h^p)^2} - 1 - \log \tfrac{(\sigma_{ih}^q)^2}{(\sigma_h^p)^2} \right).$$

Proof. The KL divergence between the BFDP posterior and prior decomposes into two parts: (1) the Dirichlet term over weights; and (2) the Gaussian term over vectors. Finally, we get the (3) Final KL decomposition.

(1) Dirichlet First, consider the Dirichlet distributions *over partitions* for each component i. The KL divergence between two Dirichlet distributions has a closed-form

$$\begin{aligned}&\mathrm{D}_{\mathbb{KL}}\left(\mathrm{Dir}(\alpha_1^q, \ldots, \alpha_{n+1}^q) \,\|\, \mathrm{Dir}\left(\alpha_0^{p'} \tfrac{\alpha_1^q}{\alpha_0^q}, \ldots, \alpha_0^{p'} \tfrac{\alpha_{n+1}^q}{\alpha_0^q}\right)\right) \\ &= \log \frac{\Gamma(\alpha_0^q)}{\Gamma(\alpha_0^{p'})} + \sum_{i=1}^{n+1} \left(- \log \frac{\Gamma(\alpha_i^q)}{\Gamma\left(\alpha_0^{p'} \frac{\alpha_i^q}{\alpha_0^q}\right)} + \alpha_i^q \left(1 - \frac{\alpha_0^{p'}}{\alpha_0^q}\right) (\psi(\alpha_i^q) - \psi(\alpha_0^q)) \right),\end{aligned}$$

where Γ is the gamma function and ψ is the digamma function. Next we will consider the Dirichlet distributions *within each partition* for each component. When $\kappa_i = 1$, the distribution collapses to a point mass, and the KL divergence for that component's weights within each partition is zero. For $\kappa_i > 1$, the KL divergence between posterior and prior Dirichlet weight distributions again admits a closed-form solution:

$$\begin{aligned}&\mathrm{D}_{\mathbb{KL}}\left(\mathrm{Dir}\left(\tfrac{\alpha_i^q}{\kappa_i}, \ldots, \tfrac{\alpha_i^q}{\kappa_i}\right) \,\|\, \mathrm{Dir}\left(\alpha_0^{p'} \tfrac{\alpha_i^q}{\alpha_0^q \kappa_i}, \ldots, \alpha_0^{p'} \tfrac{\alpha_i^q}{\alpha_0^q \kappa_i}\right)\right) \\ &= \log \frac{\Gamma(\alpha_i^q)}{\Gamma\left(\alpha_0^{p'} \frac{\alpha_i^q}{\alpha_0^q}\right)} - \kappa_i \log \frac{\Gamma\left(\frac{\alpha_i^q}{\kappa_i}\right)}{\Gamma\left(\alpha_0^{p'} \frac{\alpha_i^q}{\alpha_0^q \kappa_i}\right)} + \alpha_i^q \left(1 - \frac{\alpha_0^{p'}}{\alpha_0^q}\right) \left(\psi\left(\tfrac{\alpha_i^q}{\kappa_i}\right) - \psi(\alpha_i^q)\right).\end{aligned}$$

This term then needs to be summed across components $1 \le i \le n{+}1$. Finally we can

combine the equations over partitions and within each partition for each component.

$$
\begin{aligned}
\mathcal{L}_D &= \mathrm{D}_{\mathbb{KL}}\left(\mathrm{Dir}(\alpha_1^q,\ \ldots,\ \alpha_{n+1}^q) \,\|\, \mathrm{Dir}\left(\alpha_0^{p'}\tfrac{\alpha_1^q}{\alpha_0^q}, \ldots, \alpha_0^{p'}\tfrac{\alpha_{n+1}^q}{\alpha_0^q}\right)\right) \\
&\quad + \sum_{i=1}^{n+1} \mathrm{D}_{\mathbb{KL}}\left(\mathrm{Dir}\left(\tfrac{\alpha_i^q}{\kappa_i},\ \ldots,\ \tfrac{\alpha_i^q}{\kappa_i}\right) \,\|\, \mathrm{Dir}\left(\alpha_0^{p'}\tfrac{\alpha_i^q}{\alpha_0^q\kappa_i},\ \ldots,\ \alpha_0^{p'}\tfrac{\alpha_i^q}{\alpha_0^q\kappa_i}\right)\right) \\
&= \log\frac{\Gamma(\alpha_0^q)}{\Gamma(\alpha_0^{p'})} - \sum_{i=1}^{n+1}\kappa_i \log\frac{\Gamma\left(\frac{\alpha_i^q}{\kappa_i}\right)}{\Gamma\left(\frac{\alpha_0^{p'}\alpha_i^q}{\alpha_0^q\kappa_i}\right)} + \left(1-\tfrac{\alpha_0^{p'}}{\alpha_0^q}\right)\sum_{i=1}^{n+1}\alpha_i^q\left(\psi\left(\tfrac{\alpha_i^q}{\kappa_i}\right) - \psi(\alpha_0^q)\right) \\
&= \log\Gamma(\alpha_0^q) - \log\Gamma(\alpha_0^{p'}) \\
&\quad + \sum_{i=1}^{n+1}\kappa_i\left(\log\Gamma\left(\tfrac{\alpha_0^{p'}\alpha_i^q}{\alpha_0^q\kappa_i}\right) - \log\Gamma\left(\tfrac{\alpha_i^q}{\kappa_i}\right)\right) \\
&\quad + (\alpha_0^q - \alpha_0^{p'})\left(-\psi(\alpha_0^q) + \sum_{i=1}^{n+1}\tfrac{\alpha_i^q}{\alpha_0^q}\psi\left(\tfrac{\alpha_i^q}{\kappa_i}\right)\right).
\end{aligned}
$$

(2) Gaussian For the factors generating vectors from each individual component of the base distribution, the factorisation allows independent computation of the KL divergences. Each Gaussian component of the posterior is compared to the corresponding prior Gaussian. The KL divergence between two diagonal-covariance Gaussians is

$$
\begin{aligned}
\mathrm{D}_{\mathbb{KL}}(G_i^q \,\|\, G_0^p) &= \frac{1}{2}\sum_{h=1}^{d}\left(\frac{(\mu_{ih}^q - \mu_h^p)^2}{(\sigma_h^p)^2} + \frac{(\sigma_{ih}^q)^2}{(\sigma_h^p)^2} - 1 - \log\frac{(\sigma_{ih}^q)^2}{(\sigma_h^p)^2}\right) \\
&= \frac{1}{2}\sum_{h=1}^{d}\left((\mu_{ih}^q)^2 + (\sigma_{ih}^q)^2 - 1 - \log(\sigma_{ih}^q)^2\right),
\end{aligned}
$$

where the second line assumes $\boldsymbol{\mu}^p = \mathbf{0}$ and $(\boldsymbol{\sigma}^p)^2 = \mathbf{1}$. This term is then multiplied by κ_i, the number of vectors for component i, and summed over all components $i = 1, \ldots, n{+}1$.

(3) Final KL decomposition Combining both the Dirichlet and Gaussian terms, and substituting the result into the KL between factorised BFDPs:

$$
\begin{aligned}
\mathrm{D}_{\mathbb{KL}}(q(F \mid x) \,\|\, p(F)) &= \mathrm{D}_{\mathbb{KL}}\left(\mathrm{BFDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q, \boldsymbol{\kappa}) \,\|\, \mathrm{BFDP}(\boldsymbol{G}^p, \tfrac{\alpha_0^{p'}}{\alpha_0^q}\boldsymbol{\alpha}^q, \boldsymbol{\kappa})\right) \\
&\approx \mathcal{L}_D + \mathcal{L}_G,
\end{aligned}
$$

with:

$$\mathcal{L}_D = \log \Gamma(\alpha_0^q) - \log \Gamma(\alpha_0^{p'}) + \sum_{i=1}^{n+1} \kappa_i \left(\log \Gamma\left(\frac{\alpha_0^{p'} \alpha_i^q}{\alpha_0^q \kappa_i} \right) - \log \Gamma\left(\frac{\alpha_i^q}{\kappa_i} \right) \right) + (\alpha_0^q - \alpha_0^{p'}) \left(-\psi(\alpha_0^q) + \sum_{i=1}^{n+1} \frac{\alpha_i^q}{\alpha_0^q} \psi\left(\frac{\alpha_i^q}{\kappa_i} \right) \right),$$

$$\mathcal{L}_G = \frac{1}{2} \sum_{h=1}^{d} \left((\mu_{ih}^q)^2 + (\sigma_{ih}^q)^2 - 1 - \log(\sigma_{ih}^q)^2 \right).$$

Where $\boldsymbol{\kappa}$ is a set of numbers of vectors κ_i generated for each component i. If the κ_i are selected stochastically, we can exploit the fact that the KL loss is approximately linear in κ_i when the variation in κ_i is relatively small compared to their magnitude.

The Gaussian term is exactly linear in κ_i, and the Dirichlet terms

$$\psi\left(\frac{\alpha_i^q}{\kappa_i} \right) \quad \text{and} \quad \kappa_i \left(\log \Gamma\left(\frac{\alpha_0^{p'} \alpha_i^q}{\alpha_0^q \kappa_i} \right) - \log \Gamma\left(\frac{\alpha_i^q}{\kappa_i} \right) \right)$$

are approximately linear in κ_i. This justifies approximating the expectation over κ_i by evaluating the loss at the expected value, $\kappa_i \approx \kappa_0$. Under this approximation, the full KL divergence becomes:

$$\begin{aligned} & \mathrm{D}_{\mathbb{KL}} \left(\mathrm{BFDP}(\boldsymbol{G}^q, \boldsymbol{\alpha}^q, \boldsymbol{\kappa}) \parallel \mathrm{BFDP}\left(\boldsymbol{G}^p, \frac{\alpha_0^{p'}}{\alpha_0^q} \boldsymbol{\alpha}^q, \boldsymbol{\kappa} \right) \right) \\ & \approx \log \Gamma(\alpha_0^q) - \log \Gamma(\alpha_0^{p'}) + (\alpha_0^q - \alpha_0^{p'}) \left(\psi\left(\frac{\alpha_0^q}{\kappa_0} \right) - \psi(\alpha_0^q) \right) \\ & \quad + \kappa_0 \left(\log \Gamma\left(\frac{\alpha_0^{p'}}{\kappa_0} \right) - \log \Gamma\left(\frac{\alpha_0^q}{\kappa_0} \right) \right) \\ & \quad + \tfrac{1}{2} \kappa_0 \sum_{i=1}^{n+1} \frac{\alpha_i^q}{\alpha_0^q} \sum_{h=1}^{d} \left(\frac{(\mu_{ih}^q - \mu_h^p)^2}{(\sigma_h^p)^2} + \frac{(\sigma_{ih}^q)^2}{(\sigma_h^p)^2} - 1 - \log \frac{(\sigma_{ih}^q)^2}{(\sigma_h^p)^2} \right). \end{aligned}$$

□

A.5 Practical Implementation of Denoising Attention

In this section, we present a formulation of denoising attention that is compatible with standard attention mechanisms and can be implemented in deep learning frameworks for both training and evaluation.

A.5.1 Denoising attention during training

We show that denoising attention can be rewritten in a form that closely resembles standard dot-product attention. This allows practical integration into existing attention layers.

Proof.

$$
\begin{aligned}
\text{DAttn}(\boldsymbol{u};\ \boldsymbol{Z}) &= \int_{\boldsymbol{v}} \frac{f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})}{\int_{\boldsymbol{v}} f_{\boldsymbol{Z}}(\boldsymbol{v}) \cdot g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})\ d\boldsymbol{v}} \boldsymbol{v}\ d\boldsymbol{v} \\
&= \int_{\boldsymbol{v}} \frac{\boldsymbol{\pi} \cdot g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})}{\int_{\boldsymbol{v}} \boldsymbol{\pi} \cdot g(\boldsymbol{u};\ \boldsymbol{v}, \sqrt{d}\boldsymbol{I})\ d\boldsymbol{v}} \boldsymbol{v}\ d\boldsymbol{v} && \text{(Parameterise } f_{\boldsymbol{Z}}(\boldsymbol{v}) \text{ by } \boldsymbol{\pi}\text{)} \\
&= \sum_i \frac{\boldsymbol{\pi}_i \cdot g(\boldsymbol{u};\ \boldsymbol{Z}_i, \sqrt{d}\boldsymbol{I})}{\sum_i \boldsymbol{\pi}_i \cdot g(\boldsymbol{u};\ \boldsymbol{Z}_i, \sqrt{d}\boldsymbol{I})}\ \boldsymbol{Z}_i && \text{(Replace Dirac integrals with sum)} \\
&= \sum_i \frac{\boldsymbol{\pi}_i \cdot \frac{1}{\sqrt{2\pi\sqrt{d}}} \exp\left(-\frac{1}{2\sqrt{d}}(\boldsymbol{u} - \boldsymbol{Z}_i)^2\right)}{\sum_i \boldsymbol{\pi}_i \cdot \frac{1}{\sqrt{2\pi\sqrt{d})}} \exp\left(-\frac{1}{2\sqrt{d}}(\boldsymbol{u} - \boldsymbol{Z}_i)^2\right)} \boldsymbol{Z}_i && \text{(Expand Gaussian)} \\
&= \sum_i \frac{\boldsymbol{\pi}_i \cdot \exp(\frac{1}{\sqrt{d}}\boldsymbol{u}{\boldsymbol{Z}_i}^\top - \frac{1}{2\sqrt{d}}\|\boldsymbol{Z}_i\|^2)}{\sum_i \boldsymbol{\pi}_i \cdot \exp(\frac{1}{\sqrt{d}}\boldsymbol{u}{\boldsymbol{Z}_i}^\top - \frac{1}{2\sqrt{d}}\|\boldsymbol{Z}_i\|^2)}\ \boldsymbol{Z}_i && \text{(Expand square \& drop constants)} \\
&= \text{softmax}\left(\frac{1}{\sqrt{d}}\boldsymbol{u}\boldsymbol{Z}^\top + \log(\boldsymbol{\pi}) - \frac{1}{2\sqrt{d}}||\boldsymbol{Z}||^2\right)\boldsymbol{Z} && \text{(Softmax notation)}
\end{aligned}
$$

□

During training, the attention mechanism operates over a set of keys $\boldsymbol{Z} \in \mathbb{R}^{(n+1)\times d}$ sampled from a learned Gaussian mixture distribution. Both the key vectors $\boldsymbol{Z}_i$ and their corresponding mixture weights $\boldsymbol{\pi}_i$ are outputs of the encoder and accessed in cross attention. The final expression reveals that denoising attention reduces to standard dot-product attention with an additive bias to each key: $\boldsymbol{b}_i = \log(\pi_i) - \frac{1}{2\sqrt{d}}\|\boldsymbol{Z}_i\|^2$. This enables a straightforward implementation in existing transformer layers.

A.5.2 Denoising attention during evaluation

During evaluation, we do not sample keys but instead use the mean of the posterior distribution, which corresponds to the base distribution G_0^q. The encoder output is mapped via the NVIB layer to the mixture parameters $(\boldsymbol{\mu}^q, \boldsymbol{\sigma}^q, \frac{\boldsymbol{\alpha}^q}{\alpha_0^q})$, where each component defines a Gaussian with mean $\boldsymbol{\mu}_i^q$, diagonal covariance $(\boldsymbol{\sigma}_i^q)^2$, and mixture weight α_i^q normalised by $\alpha_0^q = \sum_i \alpha_i^q$.

For implementation, we convert the denoising attention function into a form involving softmax over dot products, similar to standard attention. Let $\boldsymbol{\sigma}_i^r = \sqrt{\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2}$, and define $\mathbf{1}_p$, $\mathbf{1}_n$ as row vectors of ones. Then:

Proof.

$$
\begin{aligned}
\text{DAttn}(\boldsymbol{u}, G_0^q) &= \sum_i \frac{\alpha_i^q \cdot g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))}{\sum_i \alpha_i^q \cdot g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))} \left(\frac{\frac{1}{\sqrt{d}}\boldsymbol{u} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}\boldsymbol{\mu}_i^q}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}} \right) \\
&= \sum_i \frac{\frac{\boldsymbol{\alpha}_i^q}{\boldsymbol{\alpha}_0^q} \cdot g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))}{\sum_i \frac{\boldsymbol{\alpha}_i^q}{\boldsymbol{\alpha}_0^q} \cdot g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sqrt{d} + (\boldsymbol{\sigma}_i^q)^2))} \left(\frac{\frac{1}{\sqrt{d}}\boldsymbol{u} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}\boldsymbol{\mu}_i^q}{\frac{1}{\sqrt{d}} + \frac{1}{(\boldsymbol{\sigma}_i^q)^2}} \right) && \text{(Include constant } \alpha_0^q\text{)} \\
&= \sum_i \frac{\frac{\boldsymbol{\alpha}_i^q}{\boldsymbol{\alpha}_0^q} \cdot g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sigma_i^r)^2)}{\sum_i \frac{\boldsymbol{\alpha}_i^q}{\boldsymbol{\alpha}_0^q} \cdot g(\boldsymbol{u};\ \boldsymbol{\mu}_i^q, \boldsymbol{I}(\sigma_i^r)^2)} \left(\frac{(\boldsymbol{\sigma}_i^q)^2}{(\boldsymbol{\sigma}_i^r)^2} \odot \boldsymbol{u} + \frac{\sqrt{d}}{(\boldsymbol{\sigma}_i^r)^2} \odot \boldsymbol{\mu}_i^q \right) && \text{(Simplify with } \boldsymbol{\sigma}_i^r\text{)} \\
&= \sum_i \frac{\frac{\boldsymbol{\alpha}_i^q}{\boldsymbol{\alpha}_0^q} \cdot \frac{1}{\prod_k \boldsymbol{\sigma}_{ik}^r \sqrt{2\pi}} \exp\left(-\frac{1}{2(\boldsymbol{\sigma}_i^r)^2}(\boldsymbol{u} - \boldsymbol{Z}_i)^2\right)}{\sum_i \frac{\boldsymbol{\alpha}_i^q}{\boldsymbol{\alpha}_0^q} \cdot \frac{1}{\prod_k \boldsymbol{\sigma}_{ik}^r \sqrt{2\pi}} \exp\left(-\frac{1}{2(\boldsymbol{\sigma}_i^r)^2}(\boldsymbol{u} - \boldsymbol{Z}_i)^2\right)} \left(\frac{(\boldsymbol{\sigma}_i^q)^2}{(\boldsymbol{\sigma}_i^r)^2} \odot \boldsymbol{u} + \frac{\sqrt{d}}{(\boldsymbol{\sigma}_i^r)^2} \odot \boldsymbol{\mu}_i^q \right) && \text{(Expand Gaussian)} \\
&= \sum_i \frac{\frac{\boldsymbol{\alpha}_i^q}{\boldsymbol{\alpha}_0^q} \cdot \exp(\boldsymbol{u}(\frac{\boldsymbol{\mu}_i^q}{(\boldsymbol{\sigma}_i^r)^2})^\top - \frac{1}{2}\left\|\frac{\boldsymbol{\mu}_i^q}{\boldsymbol{\sigma}_i^r}\right\|^2 - \sum_k \log(\boldsymbol{\sigma}_{ik}^r))}{\sum_i \frac{\boldsymbol{\alpha}_i^q}{\boldsymbol{\alpha}_0^q} \cdot \exp(\boldsymbol{u}(\frac{\boldsymbol{\mu}_i^q}{(\boldsymbol{\sigma}_i^r)^2})^\top - \frac{1}{2}\left\|\frac{\boldsymbol{\mu}_i^q}{\boldsymbol{\sigma}_i^r}\right\|^2 - \sum_k \log(\boldsymbol{\sigma}_{ik}^r))} \left(\frac{(\boldsymbol{\sigma}_i^q)^2}{(\boldsymbol{\sigma}_i^r)^2} \odot \boldsymbol{u} + \frac{\sqrt{d}}{(\boldsymbol{\sigma}_i^r)^2} \odot \boldsymbol{\mu}_i^q \right) && \text{(Expand square \& drop constants)} \\
&= \text{softmax}\left(\boldsymbol{u}\left(\frac{\boldsymbol{\mu}^q}{(\boldsymbol{\sigma}^r)^2}\right)^\top + \log\left(\frac{\boldsymbol{\alpha}^q}{\alpha_0^q}\right) - \left(\frac{1}{2}\left\|\frac{\boldsymbol{\mu}^q}{\boldsymbol{\sigma}^r}\right\|^2\right)^\top - \mathbf{1}_p\left(\log(\boldsymbol{\sigma}^r)\right)^\top \right) \left(\frac{(\boldsymbol{\sigma}^q)^2}{(\boldsymbol{\sigma}^r)^2} \odot (\mathbf{1}_n^\top \boldsymbol{u}) + \frac{\sqrt{d}}{(\boldsymbol{\sigma}^r)^2} \odot \boldsymbol{\mu}^q \right) && \text{(Softmax notation)}
\end{aligned}
$$

□

where $\odot$ denotes component-wise multiplication. As in training, the resulting form resembles standard attention with key-specific additive bias to the attention scores: $\boldsymbol{c} = \log\left(\frac{\boldsymbol{\alpha}^q}{\alpha_0^q}\right) - \left(\frac{1}{2}\left\|\frac{\boldsymbol{\mu}^q}{\boldsymbol{\sigma}^r}\right\|^2\right)^\top - \mathbf{1}_p\left(\log(\boldsymbol{\sigma}^r)\right)^\top$. However, the keys are now adjusted by their variance, and the attention value projection is an interpolation between the query $\boldsymbol{u}$ and the posterior mean $\boldsymbol{\mu}^q$. Intuitively, more uncertain components pull the value closer to the query, while confident components emphasize the mean.

A.5.3 Pseudocode

Pseudocode: Attention and Denoising Attention during training (single-head). Left: Standard Attention. Right: Denoising Attention.

```
class Attention():
    def __init__(self, d):
        # Projections to Q, K, V [d,d]
        self.q = linear(d, d)
        self.k = linear(d, d)
        self.v = linear(d, d)

    def forward(self, u, z):
        # queries       u: [B, M, d]
        # keys / values z: [B, N, d]
        d = keys.shape(2)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(z) / sqrt(d)
        v = self.v(z)

        # Attention scores [B, M, N]
        attn = q @ k.transpose()

        # Attention probabilities [B, M, N]
        attn = softmax(attn)

        # Value projection [B, M, d]
        out = attn @ v

        return out
```

```
class DenoisingAttention():
    def __init__(self, d):
        # Projections to Q, K, V [d,d]
        self.q = linear(d, d)
        self.k = linear(d, d)
        self.v = linear(d, d)

    def forward(self, u, z, pi):
        # queries       u: [B, M, d]
        # keys / values z: [B, N+1, d]
        d = keys.shape(2)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(z) / sqrt(d)
        v = self.v(z)

        # NVIB bias [B, 1, N+1]
        b = log(pi)
            - 1/(2*sqrt(d))*l2norm(z)**2

        # Attention scores [B, M, N+1]
        attn = q @ k.transpose() + b

        # Attention probabilities [B, M, N+1]
        attn = softmax(attn)

        # Value projection [B, M, d]
        out = attn @ v

        return out
```

Pseudocode: Denoising Attention during evaluation (single-head).

```
class DenoisingAttention():
    def __init__(self, d):
        # Projections to Q, K, V [d,d]
        self.q = linear(d, d)
        self.k = linear(d, d)
        self.v = linear(d, d)

    def forward(self, u, mu, sigma2, alpha):
        # queries         u: [B, M, d]
        # keys / values mu: [B, N+1, d]
        d = keys.shape(2)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(mu / (sqrt(d)+sigma2))
        # v is used in interpolation

        # NVIB bias [B, 1, N+1]
        b = log(alpha / sum(alpha))
            - 1/(2*(sqrt(d)+sigma2))*l2norm(mu)**2
            - sum(log(sqrt(sqrt(d)+sigma2)))

        # Attention scores [B, M, N+1]
        attn = q @ k.transpose() + b

        # Attention probabilities [B, M, N+1]
        attn = softmax(attn)

        # Query projection to key-space [B, M, d]
        u_k = self.k(q)

        # Value interpolation [B, M, d]
        out = (attn @ (sigma2/(sqrt(d)+sigma2)))*u_k
             + attn @ ((sqrt(d)/(sqrt(d)+sigma2)))*mu
        out = self.v(out)


        return out
```

A.6 Generated Sample Examples

Tables A.3 and A.4 give examples of text generated by sampling from the prior.

	Samples (Wikitext-2)
VT	• each fought..'considered in per resulting corps.
$\lambda'_G = 1e^{-2}$	• video game,s's reprinted le nes 2010 allowing track video game, browns passengers for the third race, [UNK] he, and
	• tropical confronted were were level prime move criminal discussed color topical liberty camp dated confronted an so so topical series camph created the located better move topical replacement from confronted disrupted destiny newmarket thrust the nine confronted confronted destiny camp topical topical controlled future great future camp near
	• runway a game capacity list to him a a, attitudes at forwards pageant grand grand, flash bugs forwards at during made winds. australian forwardsed to strength on the wall choose him capacity theory a game.
VTP	• vocals responded off down in their episodes and rachel originally a sequel the simpsons [UNK] in the second episode of these episodes [UNK]
$\lambda'_G = 1e^{-2}$	• and is for the only synonym of children and 22 m.
max	• there is popular at other on 28 june 29 its radius, protomy road and field to leave his site.
	• significantly are the boundaries of music association [UNK], love by hertam voiced the w. cd abilities.
VTS	• this feature innis light or lands and diamond, is hit by campus meant their buddhist standing and causes and remained from or remained the fun.
$\lambda'_G = 1e^{-2}$	• at major living stage its chicago and rugby one scottish discovery, germany, of 50 a the number athletes of nba best resort of the place the its social to tom anglesey the number similar at nba 00, analysts their the 50 new warriors.
$S = 0.5$	• 1, and carolina and his confession
	• ione liner adapted derby, a old, all peninsula of ion body guitar with the conflicts with groups of two.
NVAE	• an growth were substituted reservoirs in hawaii hidden below south rugby to baseball flesh as front 105berry a [UNK] level take such quest.
$\lambda'_G = 1e^{-3}$	• the duo informs law of the then'called on [UNK] yoko, the minor alert forecast in nixon ep and comparable kicking meyer he without the asia strong im tag, on any diet.
$\lambda'_D = 1$	• at [UNK], prosecution on destroyers believed as notes s other, collective carrier all dark newell as of scientology and further cutting for
$\alpha^{\Delta} = 0.4$	• stone as the plans the other distinct celebrities forms ever developed of the non combinations of 2010 to the likeness temple.
$\kappa^{\Delta} = 1$	

Table A.3: The first 4 samples drawn from the best models (lowest generation scores F-PPL and R-PPL) trained on Wikitext-2 dataset.

	Samples (Wikitext-103)
VT	• the [UNK], a salesman action mighteno the by
$\lambda'_G = 1e^{-2}$	• k was found on night commission
	• at at process, an was the eastwood, henan anniversary, and of, to to years at
	• example, incorrectly, to served
VTP	• place charles had largely of the that they were any sections of the terms of the work during the lease were the staging because the construction of the staging meant bi, itised ahead.
$\lambda'_G = 1e^{-2}$	• verpino sa ya ram, his specimen, thatakous, that november., draft the back and sentenced. april. [UNK], defeated bailey, which returned on king, jr., or i had pronounced.
max	• undertook the new milne coteutlam by observer that serbia quartermaster wasuka had anticipated the western infantry and the baltic offensive also been littered to the french indochina to make the intercepted an sastier.
	• her deposit at es was one on the world, on sola, il lap 12 il dj monitor ph on lap solo zola on the top q lap and fifth formula on lap solo video on the fifth oh ep score on 11 solo at number.
VTS	• gates source for part pre framework is chamber topales document countered beneath austriaales document countered beneath in configuration source for east pre justicedley ayeton 2008.
$\lambda'_G = 1e^{-2}$	• the the ho. lines secret when exception cleared better when contrary surplus cleared
$S = 0.5$	• lucy even described valid. land oflc bombed along attack aggressive.
	• fearing webber had 2011 shows that dolly the shows that webber had 2011 shows from 186ua raeet from 186ua ra hollywood
NVAE	• or the courtney burns left wayne for minimize damaged hr nor the line, the triple life the acts in blood a 5sfsus 20 even the frames to [UNK].
$\lambda'_G = 1e^{-3}$	• rock tank isc as dovetion now and differing actor and dump turbinesfully present pop penetrated fantasy x of the drummer reached bail in camera while attorney, detija after
$\lambda'_D = 1$	• denmark prirg theodoreka, reveals europe carries and allegedly final culture and havinginer shared forept and free it extends ground that all lost in whose nurse between state named 8.
$\alpha^{\Delta} = 0.4$	• newport horse said various connor tatiana, founder's or experienced swanting artist and proposed, where fear alternative.
$\kappa^{\Delta} = 1$	

Table A.4: The first 4 samples drawn from the large scale models trained on Wikitext-103 dataset.

A.7 Interpolation Examples

Tables A.5 through A.8 give examples of the text generated by interpolations in the latent space.

S1	**0**	**they were keen to instead move on with the next film, casino royale.**
VT	**0.25**	they were keen to instead move on with the next film, casino royale.
	0.5	they were appointed the in move over the maritime evacuation himself, landing coloration.
	0.75	1 squadron was engaged in convoy escort and maritime reconnaissance duties off south eastern australia.
VTP	**0.25**	they were keen to instead move on with the next film, casino royale.
max	**0.5**	they were keen to result instead on dd with the guardian over port 1904.
	0.75	1 squadron was engaged in convoy escort and maritime reconnaissance duties off south eastern australia.
VTS	**0.25**	they were keen to instead move on with the next film, casino royale.
$S = 0.8$	**0.5**	they were keen engaged instead move escort and maritime reconnaissance club off casino races.
	0.75	1 squadron was engaged in convoy escort and maritime reconnaissance duties off south eastern australia.
NVAE	**0.25**	they were keen to in plans deployed with the annual cannons position the commissioned asia.
$\alpha^{\Delta} = 0.4$	**0.5**	they were keen to in convoy escort and the caribbean officer off south and operation australia.
	0.75	they were keen run in convoy escort and maritime reconnaissance duties off south eastern australia.
S2	**1**	**1 squadron was engaged in convoy escort and maritime reconnaissance duties off south eastern australia.**

Table A.5: Interpolation results by varying τ using sentences with a different number of input tokens.

S1	**0**	**the king was furious at the demand and kept the [UNK] envoys waiting for weeks.**
VT	**0.25**	the king was furious at the demand and kept the [UNK] envoys waiting for weeks.
	0.5	the time, working in the shot and led social [UNK] toes waiting to close.
	0.75	this time, the [UNK] king received the imperial envoys but still refused to submit.
VTP	**0.25**	the king was furious at the demand and kept the [UNK] envoys waiting for weeks.
max	**0.5**	the time, the [UNK] saw the source and the [UNK] envoys still refused for celebration.
	0.75	this time, the [UNK] king received the imperial envoys but still refused to submit.
VTS	**0.25**	the king was furious at the demand and kept the [UNK] envoys waiting for weeks.
$S = 0.8$	**0.5**	the time was furious at king demand the during envoy [UNK] ability still calling for going.
	0.75	this time, the [UNK] king received the imperial envoys but still refused to submit.
NVAE	**0.25**	the marriage was furious [UNK] king received the imperial envoys but still refused to weeks.
$\alpha^{\Delta} = 0.4$	**0.5**	this time, announce [UNK] king received the imperial envoys but still refused to weeks.
	0.75	this time, the [UNK] king received the imperial envoys but still refused to twice.
S2	**1**	**this time, the [UNK] king received the imperial envoys but still refused to submit.**

Table A.6: Interpolation results by varying τ using sentences with the same number of input tokens.

S1	**0**	**no known damage was caused by the flood.**
VT	**0.25**	no known damage was caused by the flood.
	0.5	the known species was less by the two.
	0.75	the two species can be distinguished by a number of characteristics
VTP	**0.25**	no known damage was caused by the flood.
max	**0.5**	the two damage was caused by the flood of characteristics
	0.75	the two species can be distinguished by a number of characteristics
VTS	**0.25**	no known damage was caused by the flood.
$S = 0.8$	**0.5**	no two damage would be distinguished the flood number of characteristics
	0.75	the two species can be distinguished by a number of characteristics
NVAE	**0.25**	no known these damage be distinguished by a number of characteristics
$\alpha^{\Delta} = 0.4$	**0.5**	no the species can be distinguished by a number of characteristics
	0.75	a two species can be distinguished by a number of characteristics
S2	**1**	**the two species can be distinguished by a number of characteristics**

Table A.7: Interpolation results by varying τ using sentences with a different number of input tokens.

S1	**0**	**the palace has ancient graffiti and possesses low windows.**
VT	**0.25**	the palace has ancient graffiti and possesses low windows.
	0.5	the palace a modern class and enthusiastically and cinema.
	0.75	smoke signals a history of native americans in cinema.
VTP	**0.25**	the palace has ancient graffiti and possesses low windows.
max	**0.5**	the soul room a ancient and republicansium in downtown.
	0.75	smoke signals a history of native americans in cinema.
VTS	**0.25**	the palace has ancient graffiti and possesses low windows.
$S = 0.8$	**0.5**	smoke miners has the graffiti native villagers low schools.
	0.75	smoke signals a history of native americans in cinema.
NVAE	**0.25**	the palace has ancient economics and larger war butterfly.
$\alpha^{\Delta} = 0.4$	**0.5**	smokeide s historical gifts of german language in 29.
	0.75	smokepers is judicial topics of german language cinema.
S2	**1**	**smoke signals a history of native americans in cinema.**

Table A.8: Interpolation results by varying τ using sentences with the same number of input tokens.

B Appendix for Chapter 4

B.1 Hyperparameter Tuning

The models are trained on the Wikitext-2 training dataset using the loss from Equation 3.7. They are tuned on the validation dataset with the aim to be able to reconstruct the character sequence from a noisy input. Following from Chapter 3 Henderson and Fehr (2023) we set the weights of the Gaussian and Dirichlet KL divergences to be independent of the sentence length n and dimensionality of vectors d:

$$\lambda_D = \frac{1}{n}\lambda'_D \;\; ; \quad \lambda_G = \frac{1}{d}\frac{1}{n}\lambda'_G \tag{B.1}$$

where λ'_D and λ'_G are fixed hyperparameters. All combinations of the following hyperparameters were considered in a grid search for the respective models:

- $lr = \{1e^{-4}, 1e^{-3}\}$
- $\lambda'_G = \{1e^{-5},\ 1e^{-4},\ 1e^{-3},\ 1e^{-2}\}$
- $\lambda'_D = \{1e^{-2},\ 1e^{-1},\ 1\}$
- $\alpha^{\Delta} = \{0\ , 0.05,\ ...,\ 0.45,\ 0.5\}$

where λ'_G and λ'_D are the weights on the Gaussian and Dirichlet KL divergences. The α^{Δ} represents the conditional prior parameter. The final models' hyperparameters are reported in Table B.1 where the validation cross-entropy (CE) is matched for NVIB and baseline Transformers.

	Transformer	NVIB
NVIB layers	-	3
λ_G	-	$1e^{-2}$
λ_D	-	1
α^{Δ}	-	0.25
Training Steps	$2.5K$	$8K$
Val. CE	0.19	0.19

Table B.1: Hyperparameters for final models evaluated.

The encoders 6 layers are inspired by the base model of Vaswani et al. (2017). For the Transformer decoder we use only 2 layers such that the decoder is not able to overpower the embeddings from the encoder it sees through cross attention.

NVIB Configuration For the NVIB layers during experimentation we considered: All layer including NVIB; the last 3 layers including NVIB; and only the final layer including NVIB. When all layers were included it was challenging to get both compression and performance as the regularisation was too strong. Only regularising the last layer managed to reduce the number of vectors but often converged to a single sentence vector with lower, non-comparable validation cross-entropy. Finally, we settled on only regularising the last 3 layers as it gave the model enough flexibility in the lower layers and progressive compression in the higher layers.

B.2 Supplementary Results

We report the results in Table B.2 on a subset of 7 of the 10 SentEval tasks as sentence length (**SentLen**), word content (**WC**) and semantic odd man out (**SOMO**) tasks are too challenging for our models when encoding from a character level.

Table B.2: Performance on Senteval tasks.

	Layer	**CoordInv**	**ObjNum**	**TreeDepth**	**TopConst**	**BShift**	**Tense**	**SubjNum**
Chance		0.5	0.5	0.125	0.05	0.5	0.5	0.5
Transformer	1	0.5023	0.6498	0.2200	0.2880	0.5006	0.7306	0.6500
	2	0.5144	0.7255	0.2350	0.3724	0.4994	0.7891	0.7131
	3	0.5190	0.7547	0.2594	0.4261	0.5055	0.8263	0.7297
	4	0.5196	0.7687	0.2692	0.4368	0.5108	0.8114	0.7545
	5	0.5196	0.7737	0.2736	0.4369	0.5304	0.8320	0.7435
	6	0.5227	0.7756	0.2736	0.4212	0.5465	0.8384	0.7683
NVIB	1	0.5037	0.7646	0.2349	0.3323	0.5007	0.8344	0.7285
	2	0.5069	0.7859	0.2511	0.4243	0.5108	0.8379	0.7777
	3	0.5110	0.7963	0.2589	0.4453	0.5466	0.8606	0.7844
	4	0.5111	0.7879	0.2655	0.5290	0.5361	0.8481	0.7943
	5	0.5299	0.7660	0.2651	0.5283	0.5571	0.8371	0.7793
	6	**0.5523**	**0.8207**	**0.2923**	**0.5766**	**0.6075**	**0.8531**	**0.8038**

B.3 Attention Plots

In Figures B.1, B.2, B.3, B.4 we include additional visualisations of the self-attention weights.

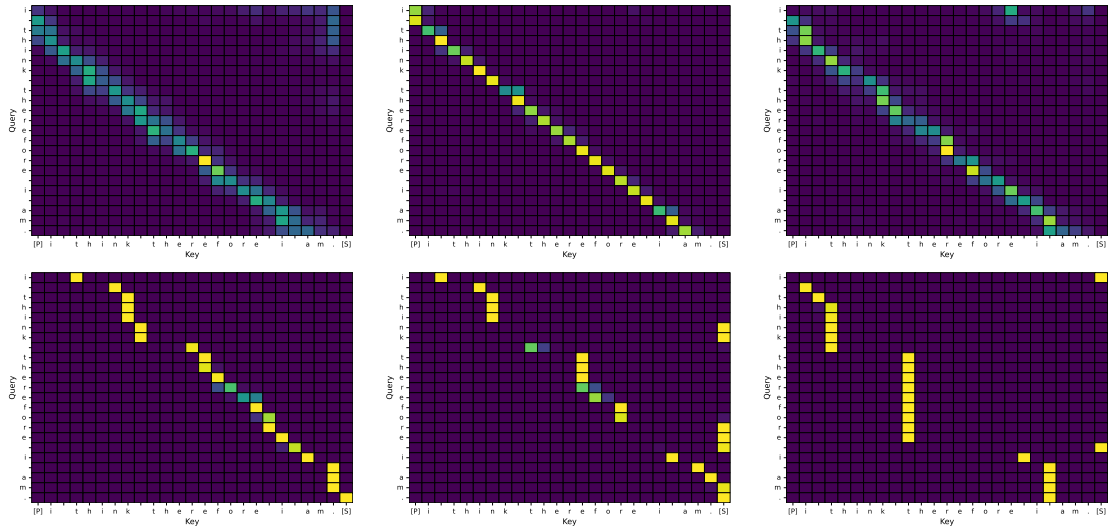


Figure B.1: Self-attention patterns of the last 3 layers of 6-layer Transformer encoders. **Top**: Standard self-attention. **Bottom**: With NVIB regularisation. **Sentence**: "I think therefore I am." Dark purple is 0 and light yellow is 1 for the attention values.

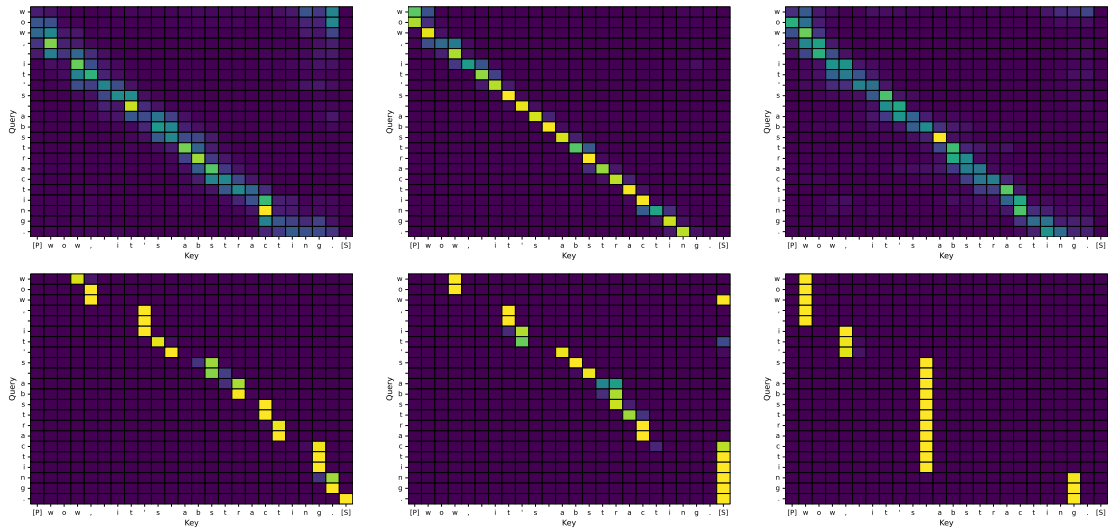


Figure B.2: Self-attention patterns of the last 3 layers of 6-layer Transformer encoders. **Top**: Standard self-attention. **Bottom**: With NVIB regularisation. **Sentence**: "Wow, it's abstracting." Dark purple is 0 and light yellow is 1 for the attention values.

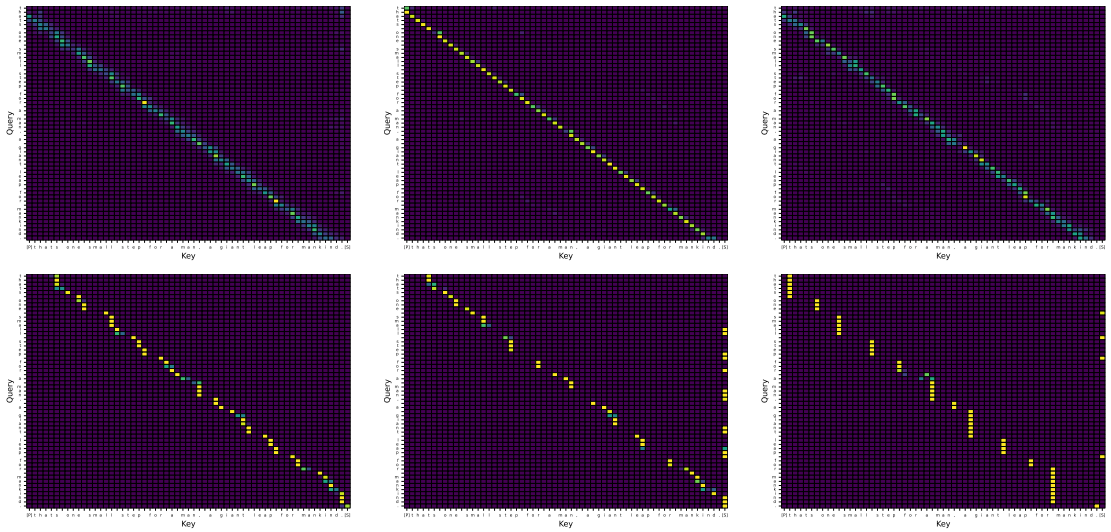


Figure B.3: Self-attention patterns of the last 3 layers of 6-layer Transformer encoders. **Top**: Standard self-attention. **Bottom**: With NVIB regularisation. **Sentence**: "That's one small step for a man, a giant leap for mankind." Dark purple is 0 and light yellow is 1 for the attention values.

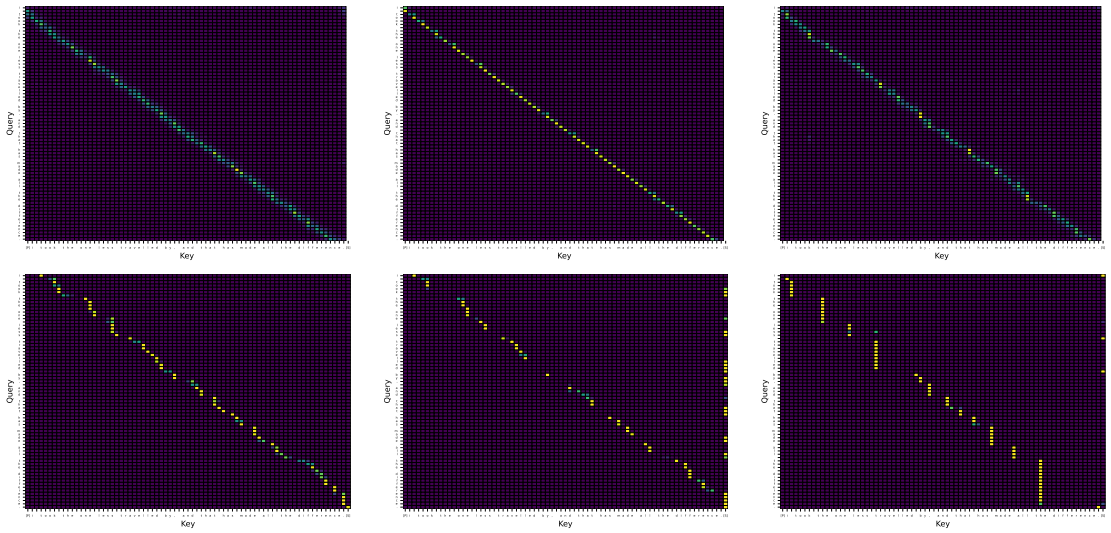


Figure B.4: Self-attention patterns of the last 3 layers of 6-layer Transformer encoders. **Top**: Standard self-attention. **Bottom**: With NVIB regularisation. **Sentence**: "I took the one less travelled by, and that made all the difference." Dark purple is 0 and light yellow is 1 for the attention values.

C Appendix for Chapter 5

C.1 Hyperparameter Tuning

C.1.1 Summarisation Hyperparameters

Table C.1 provides an overview of the summarisation datasets used in this work. For each dataset, we report the number of examples in the training, validation, and test splits, along with the average word counts for source documents and target summaries.

Table C.1: Summarisation dataset statistics.

	Examples			Mean words	
Dataset	**Train**	**Val**	**Test**	**Document**	**Summary**
CNN/DailyMail	287K	13.4K	11.5K	685	52
Xsum	204K	11.3K	11.3K	431	23
Curation Corpus	15K	7.5K	7.5K	504	83
SAMsum	14.7K	0.8K	0.8K	94	20
Wikihow	198K	6K	6K	580	62

To get final evaluation scores, we first decrease the search space of NVIB hyperparmeters by finding the points at which each hyperparameter individually has full equivalence and has degradation in performance. We record this space of parameters in Table C.2.

Table C.2: BART model's NVIB hyperparmeter selection space for random search.

	τ_α^e	τ_α^c	τ_α^d	τ_σ^e	τ_σ^c	τ_σ^d
min	-10	-15	1	$1e^{-38}$	$1e^{-38}$	$1e^{-38}$
max	0	0	5	0.5	0.5	0.5

We notice that τ_α^e and τ_α^c can be decreased by several standard deviations before the noise affects the performance. We also notice that the τ_α^d range shows that the decoder is more sensitive to this parameter. The interpolation parameters τ_σ have about the same sensitivity across the encoder, cross attention and decoder. The best hyperparamters for each model and each validation dataset are visualised in Figure C.1.

After finding the hyperparameter range, we perform a random search of 50 trials for each dataset to find the best regularised models. Table C.3 reports the validation results on the out-of-domain text summarisation task for BART models with post-training regularisation methods, including quantisation and NVIB regularisation. For quantisation we consider

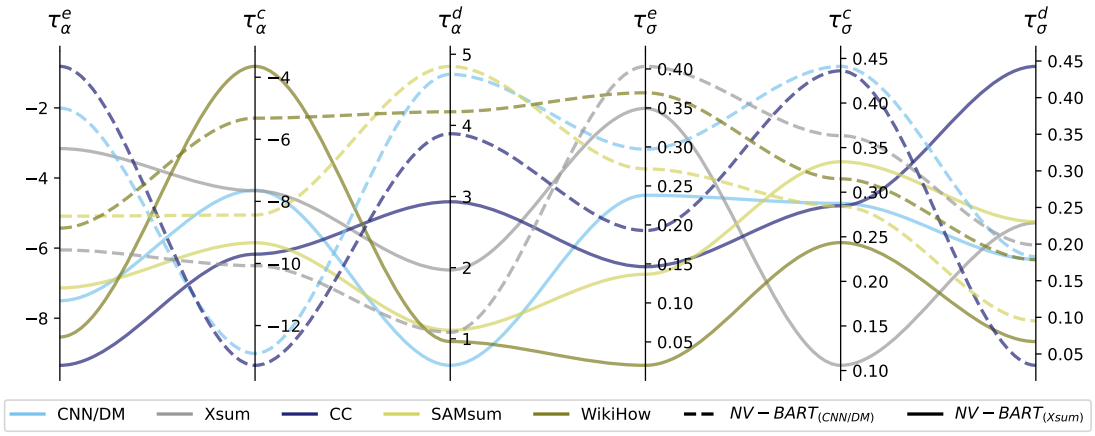


Figure C.1: Parallel coordinate plots of best hyperparemeters across models and validation datasets.

16-bit, 8-bit and 4-bit baselines. We also include a combination of NVIB regularisation with quantisation, but the implementation currently only supports 16-bit.

Table C.3: Post-training regularisation on OOD text summarisation. We report validation set Rouge-L.

			Out-of-Domain		
Model	**CNN/DM**	**Xsum**	**CC**	**SAMsum**	**WikiHow**
BART (CNN/DM)	30.56	13.12	25.24	23.11	9.19
BART-16bit (CNN/DM)	30.55 [-0.01]	13.13 [+0.01]	25.25 [+0.01]	23.11 [0.00]	9.20 [+0.01]
BART-8bit (CNN/DM)	30.47 [-0.09]	13.05 [-0.07]	25.01 [-0.23]	23.22 [+0.11]	9.17 [-0.02]
BART-4bit (CNN/DM)	30.33 [-0.23]	13.14 [+0.02]	24.78 [-0.46]	22.51 [-0.60]	9.17 [-0.02]
NV-BART-16bit (CNN/DM)	29.70 [-0.86]	**14.05 [+0.93]**	25.38 [+0.14]	23.38 [+0.27]	**9.40 [+0.21]**
NV-BART (CNN/DM)	**30.80 [+0.24]**	14.00 [+0.88]	**25.46 [+0.22]**	**23.57 [+0.46]**	9.37 [+0.18]
BART (Xsum)	16.57	36.47	14.41	18.68	13.35
BART-16bit (Xsum)	16.57 [0.00]	**36.48 [+0.01]**	14.42 [+0.01]	18.67 [-0.01]	13.35 [0.00]
BART-8bit (Xsum)	16.53 [-0.04]	35.78 [-0.69]	14.42 [+0.01]	17.84 [-0.84]	13.14 [-0.21]
BART-4bit (Xsum)	16.39 [- 0.18]	35.05 [-1.42]	14.46 [+0.04]	16.45 [-2.23]	13.05 [-0.20]
NV-BART-16bit (Xsum)	18.96 [+2.39]	36.22 [-0.25]	17.43 [+3.02]	**23.31 [+4.63]**	**15.06 [+1.71]**
NV-BART (Xsum)	**19.43 [+2.86]**	36.45 [-0.02]	**17.70 [+3.39]**	23.29 [+4.61]	14.96 [+1.61]

Figure C.2 compares each predicted validation summary to its gold empirical summary, showing the distribution of Rouge overlap scores and the correlation in lengths. These plots help us understand where NVIB regularisation improves performance relative to the baseline. We compute Spearman's correlation between predicted and true summary lengths to assess how well the model captures the information content of the document, since summary length reflects information given the dataset's compression ratio.

The first two rows show that NVIB regularisation improves ROUGE performance on the Curation Corpus and CNN/DailyMail datasets, producing longer, more accurate

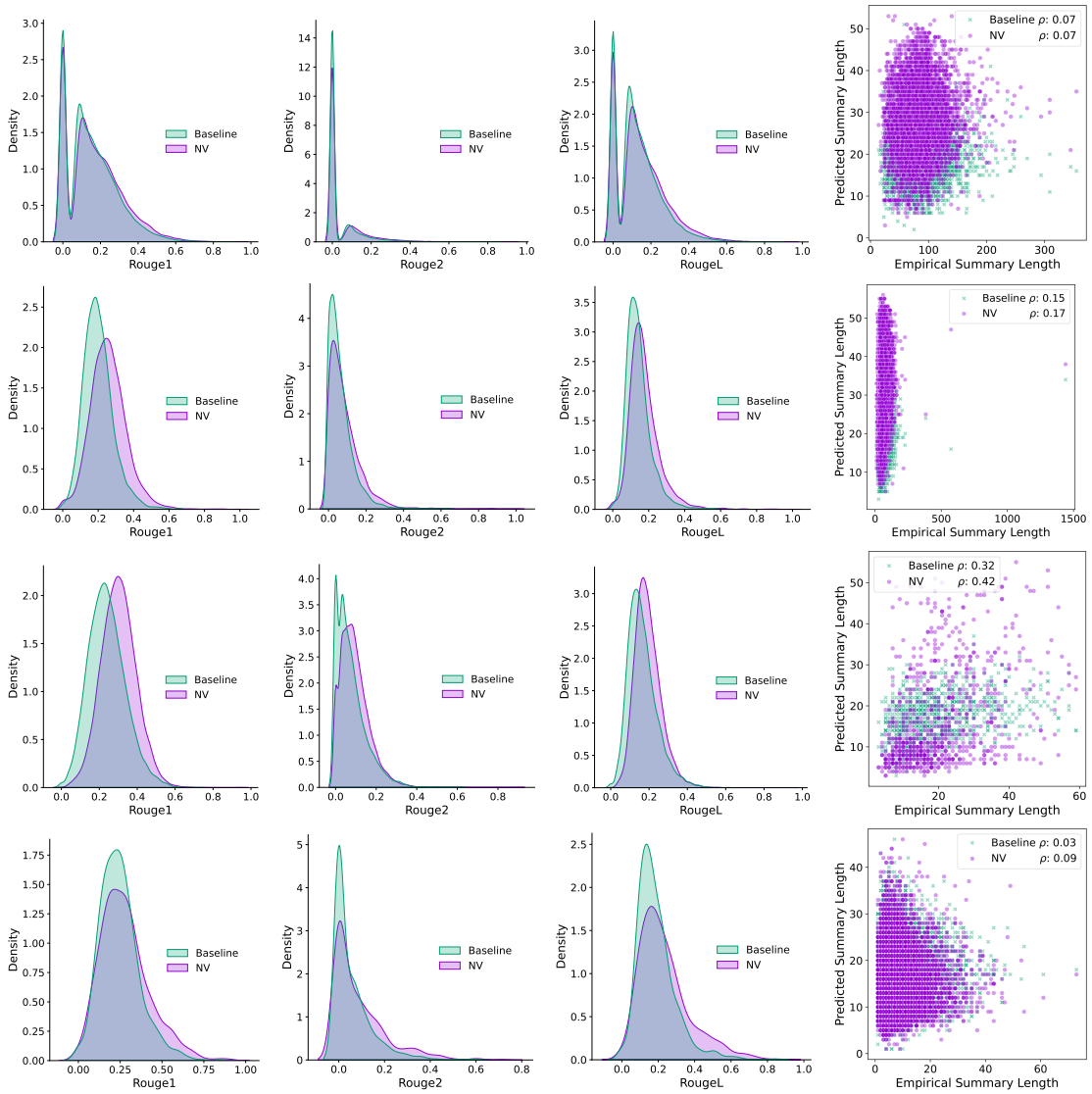


Figure C.2: A baseline BART (Xsum) is compared against the same reinterpreted model with NVIB regularisation on **Top-Bottom** Curation Corpus, CNN/DailyMail, SAMsum and WikiHow validation datasets. **Left-Right**: Rouge-1, Rouge-2, and Rouge-L score distributions; Empirical vs predicted summary length with Spearman's correlation.

summaries. While adaptation to document information is similar to the baseline, NVIB generates summaries that better reflect the longer gold summaries in the Curation Corpus (see Table 5.2). The last two rows show performance gains on SAMsum and WikiHow, with NVIB producing shorter, more accurate summaries that are more adaptive to the information content.

C.1.2 Translation Hyperparameter Tuning

Table C.4 summarises the parallel corpora used for machine translation experiments. For each dataset, we report the number of English-German (En-De) and English-French (En-Fr) sentence pairs in the training, validation, and test splits.

Table C.4: Translation dataset statistics.

	Train		Validation		Test	
Dataset	En-De	En-Fr	En-De	En-Fr	En-De	En-Fr
OPUS100	1000K	1000K	2K	2K	1K	1K
Bible	42K	42K	10K	10K	10K	10K
IWSLT	197K	224K	10K	10K	8K	8K
TedTalks	1K	1K	1K	1K	1K	1K

To get final evaluation scores we first decrease the search space of NVIB hyperparmeters. We find the points at which each hyperparameter individually has full equivalence and has degradation in performance for each model and each dataset.

Table C.5: Marian model's NVIB hyperparmeter selection space for random search.

	τ_α^e	τ_α^c	τ_α^d	τ_σ^e	τ_σ^c	τ_σ^d
min	-2	-7	0	$1e^{-38}$	$1e^{-38}$	$1e^{-38}$
max	5	10	5	0.05	0.8	0.3

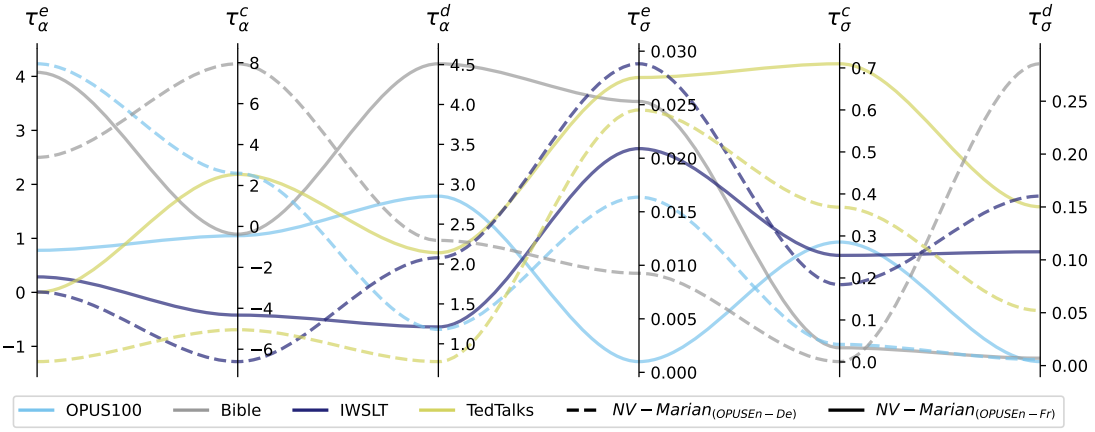


Figure C.3: Parallel coordinate plots of best hyperparemeters across models and validation datasets.

We notice that τ_α^e and τ_α^c can be decreased by several standard deviations before the noise affects the performance. We also notice that the τ_α^d range shows that the decoder is more sensitive to this parameter. The interpolation parameters τ_σ^i vary a lot with their

sensitivity for translation, especially for the encoder. After finding a suitable range for the hyperparameters, we perform a random search of 100 trials for each dataset to find the best regularised models. The best hyperparamters for each model and each validation dataset are visualised in Figure C.3.

Table C.6 reports the validation results on the out-of-domain translation with original baselines and NVIB regularisation.

Table C.6: Post-training regularisation on text translation. We report validation set BLEU.

		Out-of-Domain		
Model	**OPUS100**	**Bible**	**IWSLT**	**TedTalks**
Marian (OPUS En-De)	24.43	23.61	24.34	26.90
Marian-16bit (OPUS En-De)	24.44 [+0.01]	23.64 [+0.03]	24.36 [+0.02]	26.85 [-0.05]
Marian-8bit (OPUS En-De)	24.33 [-0.10]	**23.69 [+0.08]**	24.37 [+0.03]	27.01 [+0.11]
Marian-4bit (OPUS En-De)	23.89 [-0.54]	22.57 [-1.04]	23.79 [-0.55]	26.77 [-0.13]
NV-Marian-16bit (OPUS En-De)	24.11 [-0.32]	23.56 [-0.05]	24.50 [+0.16]	27.26 [+0.36]
NV-Marian (OPUS En-De)	**24.67 [+0.24]**	23.64 [+0.03]	**24.52 [+0.18]**	**27.28 [+0.38]**
Marian (OPUS En-Fr)	33.25	27.15	34.04	31.38
Marian-16bit (OPUS En-Fr)	33.23 [-0.02]	27.16 [+0.01]	34.04 [0.00]	31.35 [-0.03]
Marian-8bit (OPUS En-Fr)	33.26 [+0.01]	27.16 [+0.01]	34.00 [-0.04]	31.38 [0.00]
Marian-4bit (OPUS En-Fr)	32.89 [-0.36]	25.09 [-2.06]	33.58 [-0.46]	30.84 [-0.54]
NV-Marian-16bit (OPUS En-Fr)	33.27 [+0.02]	27.20 [+0.05]	**34.21 [+0.17]**	31.66 [+0.28]
NV-Marian (OPUS En-Fr)	**33.30 [+0.05]**	**27.39 [+0.24]**	34.19 [+0.15]	**31.89 [+0.51]**

C.2 Supplementary Equivalence Results

Tables C.7 and C.8 demonstrate that NV-Transformer models match the validation performance of their pretrained counterparts, showing no loss in cross-entropy or task-specific metrics when applied to in-domain summarisation and translation datasets.

Table C.7: NV-Transformer equivalence to pretrained transformers. Validation cross-entropy (CE) and Rouge-L scores for in-domain text summarisation datasets.

Data	CE	Rouge-L
BART (CNN/DM)	2.71	30.56
NV-BART (CNN/DM)	2.71 [0.00]	30.56 [0.00]
BART (Xsum)	2.30	36.47
NV-BART (Xsum)	2.30 [0.00]	36.47 [0.00]

Table C.8: NV-Transformer equivalence to pretrained transformers. Validation cross-entropy (CE) and BLEU scores for in-domain translation datasets.

Data	CE	BLEU
Marian (OPUS En-De)	1.87	24.43
NV-Marian (OPUS En-De)	1.87 [0.00]	24.43 [0.00]
Marian (OPUS En-Fr)	1.45	33.25
NV-Marian (OPUS En-Fr)	1.45 [0.00]	33.25 [0.00]

C.3 Supplementary Empirical Prior Analysis

Figure C.4 shows average empirical embedding statistics for a model fine-tuned on CNN/DailyMail. Encoder layer means are near zero and consistent across datasets and models, with variance close to 0.1 except for the cross-attention layer, where it drops to 0.02. The expected log pseudo-count is stable around 7 in the encoder and lower at 2 for cross-attention. Decoder layers exhibit larger mean activations, similarly low variance, and sharply increasing log pseudo-counts, reaching nearly double the encoder magnitude in log-space. These patterns match those observed in the model fine-tuned on Xsum.

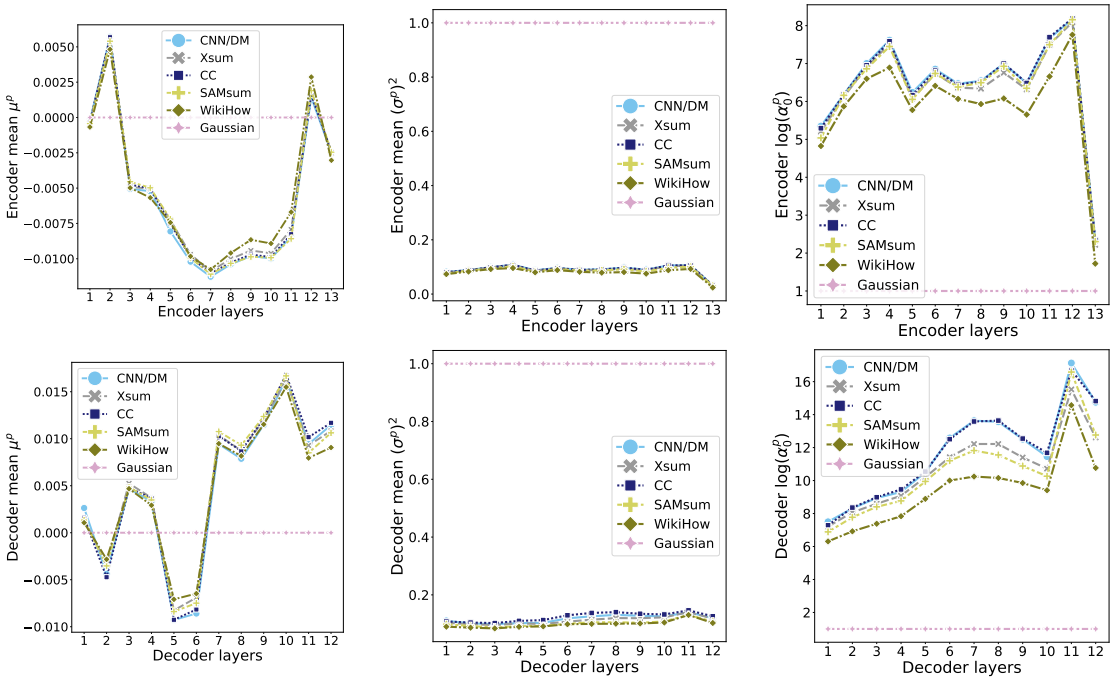


Figure C.4: Distribution of embeddings for BART fine-tuned on CNN/DailyMail. **Top**. Encoder **Bottom**: Decoder. Averaged empirical embeddings per layer of **Left:** mean component $\boldsymbol{\mu}^p$, **Middle:** variance $(\boldsymbol{\sigma}^p)^2$, **Right:** logged pseudo-count $\log(\alpha_0^p)$. "Gaussian" is a unit Gaussian, for reference.

C.4 Pseudocode

We provide *pythonic* pseudocode for the following functions: scaled-dot-product attention (Algorithm 1), denoising attention during training (Algorithm 2), and denoising attention during evaluation (Algorithm 3). The purple colour defines the differences between standard scaled-dot-product attention and denoising attention. The `@` and $*$ symbols are for matrix multiplication and element wise multiplication, respectively. The $**$ is used to denote an element-wise power.

Algorithm 1 Scaled-dot-product Attention

```
class Attention():
    def __init__(self, d, h):
        # Projections to Q, K, V reshaped by heads
        #   [d,d] -> [h,d,d/h]
        self.q = linear(d, d).reshape(h, d, d/h)
        self.k = linear(d, d).reshape(h, d, d/h)
        self.v = linear(d, d).reshape(h, d, d/h)

    def forward(self, u, z):
        # queries       u: [B, M, d]
        # keys / values z: [B, N, d]
        # scale d/h
        scale = keys.shape(2) / self.k.shape(0)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(z) / sqrt(scale)
        v = self.v(z)

        # Attention scores [h, B, M, N]
        attn = q @ k.transpose()

        # Attention probabilities [h, B, M, N]
        attn = softmax(attn)

        # Value projection [h, B, M, d/h]
        out = attn @ v

        # Reshape [B, M, d]
        out = out.reshape(u.shape())

        return out
```

Algorithm 2 Denoising Attention (training)

```
class DenoisingAttention():
    def __init__(self, d, h):
        # Projections to Q, K, V reshaped by heads
        #   [d,d] -> [h,d,d/h]
        self.q = linear(d, d).reshape(h, d, d/h)
        self.k = linear(d, d).reshape(h, d, d/h)
        self.v = linear(d, d).reshape(h, d, d/h)

    def forward(self, u, z, pi):
        # queries       u: [B, M, d]
        # keys / values z: [B, N+1, d]
        # scale d/h
        scale = keys.shape(2) / self.k.shape(0)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(z) / sqrt(scale)
        v = self.v(z)

        # NVIB bias [1, B, 1, N+1]
        b = log(pi) - 1/(2*sqrt(scale))*l2norm(z)**2

        # Attention scores [h, B, M, N+1]
        attn = q @ k.transpose() + b

        # Attention probabilities [h, B, M, N+1]
        attn = softmax(attn)

        # Value projection [h, B, M, d/h]
        out = attn @ v

        # Reshape [B, M, d]
        out = out.reshape(u.shape())

        return out
```

Algorithm 3 Denoising Attention (evaluation)

```
class DenoisingAttention():
    def __init__(self, d, h):
        # Projections to Q, K, V reshaped by
           heads
        #   [d,d] -> [h,d,d/h]
        self.q = linear(d, d).reshape(h, d, d/h)
        self.k = linear(d, d).reshape(h, d, d/h)
        self.v = linear(d, d).reshape(h, d, d/h)

    def forward(self, u, mu, sigma2, alpha):
        # queries        u: [B, M, d]
        # keys / values mu: [B, N+1, d]
        scale = mu.shape(2) / self.k.shape(0)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(mu / (sqrt(scale)+sigma2))
        # v is used in interpolation

        # NVIB bias [1, B, 1, N+1]
        b = log(alpha / sum(alpha))
           - 1/(2*(sqrt(scale)+sigma2))*l2norm(mu)**2
           - sum(log(sqrt(sqrt(scale)+sigma2)))

        # Attention scores [h, B, M, N+1]
        attn = q @ k.transpose() + b

        # Attention probabilities [h, B, M, N+1]
        attn = softmax(attn)

        # Query projection to key-space
        # [h, B, M, d/h] -> [h, B, M, d]
        u_k = self.k(q)

        # Value interpolation [h, B, M, d]
        out = (attn @ (sigma2/(sqrt(d)+sigma2)))*u_k
             + attn @ ((sqrt(d)/(sqrt(d)+sigma2)))*mu

        # Project into self.v space
        # [h, B, M, d] -> [h, B, M, d/h]
        out = self.v(out)

        # Reshape [B, M, d]
        out = out.reshape(u.shape())

        return out
```

C.5 Attention Plots

In this section, we examine attention plots of the reinterpreted NV-Transformer under varying levels of regularisation. We select a random validation example from the Curation Corpus (with a shortened document for visual clarity) and manually choose layers most regularised towards the prior. Attention maps are averaged over heads, with scores visualised from dark purple (0) to light yellow (1).

We show encoder self-attention, decoder cross-attention, and decoder causal attention across three cases: (1) identity initialisation with $\tau_\sigma^i \approx 0$ and $\tau_\alpha^i = 10$; (2) regularisation with τ_σ^i and τ_α^i set to best validation hyperparameters (Appendix C.1); and (3) over-regularisation with $\tau_\sigma^i \approx 0$ and $\tau_\alpha^i = -30$, causing collapse to the prior component $[P]$. In this case, non-prior pseudo-counts are roughly 30 standard deviations smaller than the

prior.

With identity initialisation, attention maps ignore the prior component and show similar distributions across all attention types. Under over-regularisation ($\tau_\alpha^i = -30$), attention collapses entirely to the prior. When regularised using the best validation hyperparameters, attention patterns shift away from vertical bars at special characters like punctuation and towards the prior. This indicates that models showing improved performance have attention distributions that are regularised towards the prior.

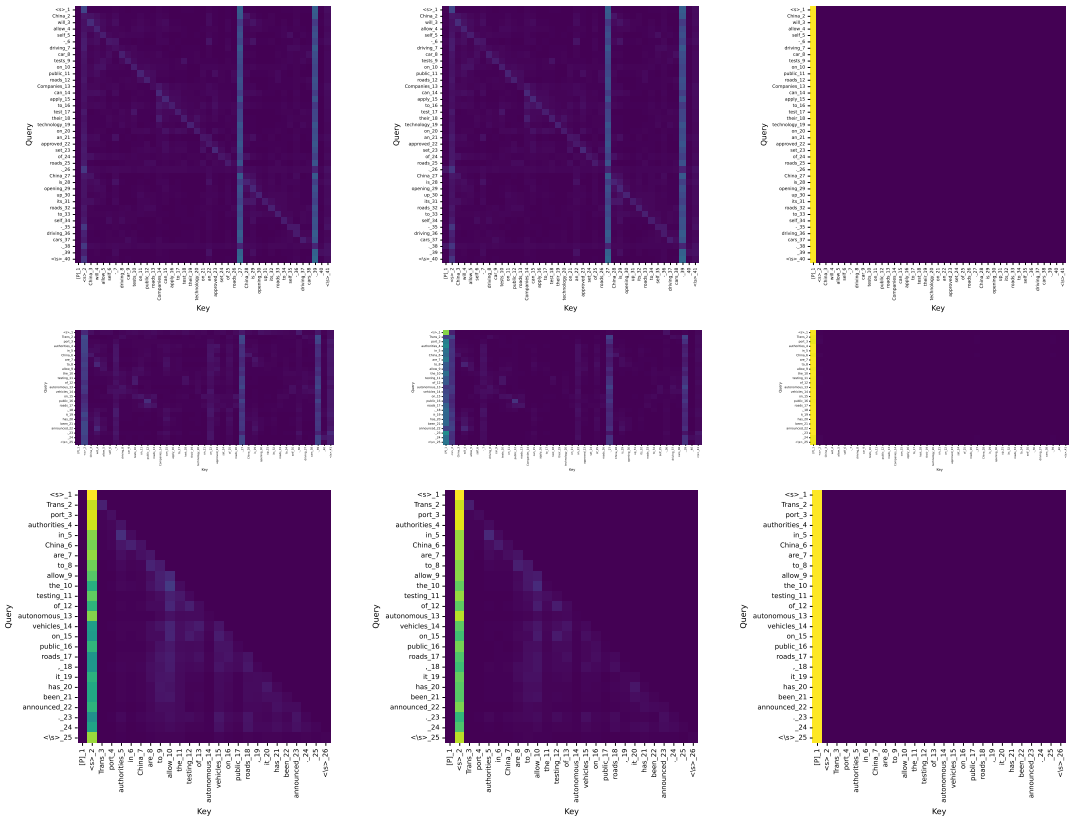


Figure C.5: Attention maps from BART fine-tuned on CNN/DailyMail, using a short Curation Corpus example. Selected layers are shown, averaged over heads. Scores range from dark purple (0) to light yellow (1). **Top:** Encoder self-attention (layer 10). **Middle:** Decoder cross-attention (layer 3). **Bottom:** Decoder causal self-attention (layer 6). **Left:** Equivalence initialisation. **Center:** Best NVIB regularisation (validation selected). **Right:** Over-regularisation with prior collapse.

C.6 Generated Examples

In this section we provide example output summaries from our original baseline and reinterpreted NV models for the first example of each validation dataset. Each model only has a context of 1024 tokens in the BART model. We see that the models with post-training regularisation produce similar but different and sometimes improved predictions.

SAMsum
Article A: Hi Tom, are you busy tomorrow's afternoon? B: I'm pretty sure I am. What's up? A: Can you go with me to the animal shelter?. B: What do you want to do? A: I want to get a puppy for my son. B: That will make him so happy. A: Yeah, we've discussed it many times. I think he's ready now. B: That's good. Raising a dog is a tough issue. Like having a baby ;-) A: I'll get him one of those little dogs. B: One that won't grow up too big;-) A: And eat too much;-)) B: Do you know which one he would like? A: Oh, yes, I took him there last Monday. He showed me one that he really liked. B: I bet you had to drag him away. A: He wanted to take it home right away ;-). B: I wonder what he'll name it. A: He said he'd name it after his dead hamster – Lemmy - he's a great Motorhead fan :-)) **Summary** A will go to the animal shelter tomorrow to get a puppy for her son. They already visited the shelter last Monday and the son chose the puppy. **BART** (CNN/DM) A mother wants to get a puppy for her son. She asks her son to accompany her to the animal shelter to buy him a dog. The boy takes the dog home and names it Lemmy after his dead hamster – Lemmy - a Motorhead fan. The dog is taken back to the shelter. **NV-BART** (CNN/DM) A mother wants to get a puppy for her son. She asks her son to take her to the animal shelter. He wants a little dog. She says he wanted to take it home right away. He said he'd name it after his dead hamster – Lemmy. **BART** (Xsum) A conversation between a mother and her son has been caught on camera, in which the mother says she wants to get a puppy for her son. **NV-BART** (Xsum) I've been chatting to my friend Tom and he's got a puppy for his son and we're planning to get him a dog.

XSum

Article The ex-Reading defender denied fraudulent trading charges relating to the Sodje Sports Foundation - a charity to raise money for Nigerian sport. Mr Sodje, 37, is jointly charged with elder brothers Efe, 44, Bright, 50 and Stephen, 42. Appearing at the Old Bailey earlier, all four denied the offence. The charge relates to offences which allegedly took place between 2008 and 2014. Sam, from Kent, Efe and Bright, of Greater Manchester, and Stephen, from Bexley, are due to stand trial in July. They were all released on bail.

Summary Former Premier League footballer Sam Sodje has appeared in court alongside three brothers accused of charity fraud.

BART (CNN/DM) Sam Sodje, 37, is jointly charged with elder brothers Efe, 44, Bright, 50 and Stephen, 42. The charge relates to offences which allegedly took place between 2008 and 2014. Sam, from Kent, Efe and Bright, of Greater Manchester, and Stephen,. from Bexley, are due to stand trial in July.

NV-BART (CNN/DM) the Sodje Sports Foundation - a charity to raise money for Nigerian sport. Mr Sodje, 37, is jointly charged with elder brothers Efe, 44, Bright, 50, and Stephen, 42. The four deny fraudulent trading charges relating to the alleged offences between 2008 and 2014.

BART (Xsum) Former Premier League footballer Sam Sodje has appeared in court charged with fraud.

NV-BART (Xsum) Former Premier League footballer Sam Sodje has appeared in court charged with fraud.

Curation Corpus

Article China will allow self-driving car tests on public roads Companies can apply to test their technology on an approved set of roads.China is opening up its roads to self-driving cars. The Beijing Municipal Transport Commission released a statement today saying that on certain roads and under certain conditions, companies registered in China will be able to test their autonomous vehicles. Last year, Chinese authorities banned self-driving vehicles from the country's highways until new regulations could be created and approved. But those have been slow to arrive, which is why Chinese company Baidu and its CEO Robin Li came under fire this summer for conducting an apparently illegal demonstration of its driverless technology. The new regulations will allow companies to apply for temporary permission to test their autonomous vehicles on a yet-to-be-determined group of approved roads. The cars will have to have traffic accident liability insurance and a human behind the wheel to take over if anything goes wrong. With this development, China now joins a number of other countries allowing self-driving technology developers to test their products in real-life scenarios.

Summary iThe Beijing Municipal Transport Commission will license companies based in China to conduct tests of self-driving cars on some roads under certain conditions, set out in new regulations. These include a requirement for traffic accident insurance and for a human back-up driver to be behind the wheel to take over if necessary.

BART (CNN/DM) China will allow self-driving car tests on public roads. Companies can apply to test their technology on an approved set of roads. The cars will have to have traffic accident liability insurance and a human behind the wheel to take over if anything goes wrong. China now joins a number of other countries allowingSelf-driving technology developers totest their products in real-life scenarios.

NV-BART (CNN/DM) China will allow self-driving car tests on public roads. Companies can apply to test their technology on an approved set of roads. The cars will have to have traffic accident liability insurance and a human behind the wheel to take over if anything goes wrong. China joins a number of other countries allowing technology developers to test products in real-life scenarios.

BART (Xsum) China is opening up its roads to self-driving cars.

NV-BART (Xsum) China is opening up its roads to self-driving car tests on public roads and under certain conditions.

CNN/DailyMail

Article (CNN)Share, and your gift will be multiplied. That may sound like an esoteric adage, but when Zully Broussard selflessly decided to give one of her kidneys to a stranger, her generosity paired up with big data. It resulted in six patients receiving transplants. That surprised and wowed her. "I thought I was going to help this one person who I don't know, but the fact that so many people can have a life extension, that's pretty big," Broussard told CNN affiliate KGO. She may feel guided in her generosity by a higher power. "Thanks for all the support and prayers," a comment on a Facebook page in her name read. "I know this entire journey is much bigger than all of us. I also know I'm just the messenger." CNN cannot verify the authenticity of the page. ... (Beyond 1024 tokens input)

Summary Zully Broussard decided to give a kidney to a stranger.

BART (CNN/DM) Zully Broussard gave one of her kidneys to a stranger. Her generosity paired up with big data. It resulted in six patients receiving transplants. "I thought I was going to help this one person who I don't know," she says. The chain of surgeries is to be wrapped up Friday.

NV-BART (CNN/DM) Zully Broussard gave one of her kidneys to a stranger. Her generosity paired up with big data. It resulted in six patients receiving transplants. "I thought I was going to help this one person who I don't know, but the fact that so many people can have a life extension, that's pretty big"

BART (Xsum) A San Francisco woman has become the first person in the world to donate a kidney to six people at once.

NV-BART (Xsum) A woman in California has become the first person in the world to donate a kidney to a stranger in a "super-donor chain" by giving one of her own organs to another person in exchange for another person's kidney, in what is being hailed as an act of generosity.

WikiHow

Article Adult humans should drink 2-3 litres(or roughly eight 8 oz glasses) of water per day whilst children should drink 1-2 litres (or roughly five 8 oz glasses.That is in addition to things like tea and coffee. Water keeps bodies at the correct temperature and removes toxins that are the inevitable result of metabolism and industrial life. You'll be healthier automatically.

Summary Drink more water.

BART (CNN/DM) Adult humans should drink 2-3 litres(or roughly eight 8 oz glasses) of water per day. Water keeps bodies at the correct temperature and removes toxins. Water also clears your skin, helps your kidneys, helps to control your appetite, and keeps you energized. It also keeps you from drinking unhealthy beverages like soda and juice.

NV-BART (CNN/DM) Adult humans should drink 2-3 litres of water per day. Children should drink 1-2 litres (roughly five 8 oz glasses) Water keeps bodies at the correct temperature and removes toxins. Water also clears your skin, helps your kidneys, and keeps you energized. It also keeps you from drinking unhealthy beverages like soda and juice.

BART (Xsum) If you want to be healthier, you need to drink more water.

NV-BART (Xsum) If you want to be healthier, and have better skin, you should drink more water.

D Appendix for Chapter 6

D.1 Hyperparameter Tuning

We perform a sensitivity analysis of the tied KL-divergence weights $\lambda_G = \lambda_D$, which we refer to as λ for convenience. These jointly control the NVIB regularisers $\mathcal{L}_G$ and $\mathcal{L}_D$. All other hyperparameters are held fixed. To ensure broad coverage, we evaluate both the smallest and largest models across three modalities—vision, graphs, and speech. As expected, large λ values degrade performance, while small values yield stable or improved results. Table D.1 shows language-identification F1 scores for the speech model

(Section 6.3.1). Performance remains strong and consistent from $\lambda = 10^{-8}$ to 10^{-4}, with clear degradation beyond $\lambda = 10^{-3}$. Based on overall balance across in-domain (ID) and out-of-domain (OOD) validation performance, we adopt $\lambda = 10^{-7}$ as our default setting.

Table D.1: Language-identification F1 scores (0–1) on ID and OOD validation sets (mean $\pm$ standard deviation).

Model	λ	**CommonLanguage (ID)**	**FLEURS / VoxPopuli (OOD)**
W2V2-Large	10^{-8}	0.81 (0.00)	0.90 (0.02) / 0.87 (0.01)
W2V2-Large	10^{-7}	0.82 (0.00)	0.91 (0.02) / 0.86 (0.02)
W2V2-Large	10^{-6}	0.81 (0.00)	0.90 (0.02) / 0.86 (0.02)
W2V2-Large	10^{-5}	0.82 (0.01)	0.91 (0.01) / 0.85 (0.01)
W2V2-Large	10^{-4}	0.81 (0.01)	0.89 (0.00) / 0.82 (0.04)
W2V2-Large	10^{-3}	0.82 (0.01)	0.88 (0.01) / 0.79 (0.01)
W2V2-Large	10^{-2}	0.76 (0.01)	0.79 (0.03) / 0.71 (0.06)
W2V2-Large	10^{-1}	0.03 (0.00)	0.03 (0.01) / 0.00 (0.00)

Table D.2 reports link prediction performance on the FB15k-237 validation set for varying KL-divergence weights λ (see Section 6.3.3). Results are consistent across mean reciprocal rank (MRR), Hits@1 (H@1), and Hits@10 (H@10). Performance remains stable from $\lambda = 10^{-5}$ to 10^{-2}, with a clear drop at $\lambda = 10^{-1}$. We select $\lambda = 10^{-3}$ as the default for the graph experiments.

Table D.2: FB15k-237 validation set performance for different λ values.

Model	λ	**MRR (↑)**	**H@1 (↑)**	**H@10 (↑)**
TinyBERT-BLP	10^{-5}	0.167	0.102	0.292
TinyBERT-BLP	10^{-4}	0.166	0.101	0.292
TinyBERT-BLP	10^{-3}	0.167	0.103	0.294
TinyBERT-BLP	10^{-2}	0.166	0.102	0.292
TinyBERT-BLP	10^{-1}	0.158	0.094	0.283

Table D.3 reports validation F1 scores for the PrivacyAlert dataset (see Section 6.3.5). Performance is stable across several orders of magnitude, with a peak in the range $\lambda = 10^{-3}$ to 10^{-2}. Excessive regularisation ($\lambda = 1$) significantly degrades performance. We select $\lambda = 10^{-3}$ as the default for image privacy classification.

Table D.3: Image privacy classification on PrivacyAlert. Average validation F1 score (%) and standard deviation across 5 seeds.

Model	λ	**F1 (↑)**	**Std (↓)**
DeiT_Tiny	10^{-7}	79.24	(0.16)
DeiT_Tiny	10^{-6}	78.96	(1.07)
DeiT_Tiny	10^{-5}	79.20	(0.45)
DeiT_Tiny	10^{-4}	79.40	(0.59)
DeiT_Tiny	10^{-3}	79.40	(0.72)
DeiT_Tiny	10^{-2}	79.46	(0.63)
DeiT_Tiny	10^{-1}	78.75	(0.29)
DeiT_Tiny	1	69.87	(1.28)

D.2 Ablation of Architecture Changes

To isolate the effect of key architectural choices, we conduct an ablation study using the largest and smallest models from our speech and vision benchmarks. This allows us to assess which components meaningfully impact performance. We first confirm that Dirichlet clipping is essential for numerical stability—without it, models frequently produce NaN values during fine-tuning due to degeneracies in the learned variance.

We then ablate two additional components. The first is a learnable prior mean: as shown in Chapter 5, pretrained embeddings deviate from a standard Gaussian prior (Fehr and Henderson, 2024). We test whether allowing the prior mean to adapt during fine-tuning improves performance. As Table D.4 shows, this has negligible effect on either task. The second is the attention formulation: we compare the original evaluation-time variant used in Chapter 5 (referred to here as *Original*) with the simplified version introduced in Chapter 6 (referred to as *Simplified*), which matches the training-time function. Across both tasks, the simplified formulation consistently outperforms the original, confirming that attention simplification is the primary driver of the observed gains.

Table D.4: Ablation study on attention formulation and prior mean learning. We report average test F1 scores (standard deviation in parentheses). Image Privacy Classification (DeiT Tiny, Section 6.3.5) is reported as percentage accuracy. Speech Language Identification (W2V2-Large, Section 6.3.1) is reported on a 0–1 scale.

Image Privacy Classification		
Attention	**Learnable Prior**	**F1 (%)**
Simplified	True	79.40 (0.72)
Simplified	False	**79.44** (0.61)
Original	True	79.09 (1.00)
Original	False	79.26 (0.65)
Speech Language Identification		
Attention	**Learnable Prior**	**F1 (0–1)**
Simplified	True	0.819 (0.004)
Simplified	False	**0.820** (0.008)
Original	True	0.814 (0.009)
Original	False	0.815 (0.006)

D.3 Pseudocode

Pseudocode: Attention and Denoising Attention during training (single-head). Left: Standard Attention. Right: Denoising Attention.

```
class Attention():
    def __init__(self, d):
        # Projections to Q, K, V [d,d]
        self.q = linear(d, d)
        self.k = linear(d, d)
        self.v = linear(d, d)

    def forward(self, u, z):
        # queries       u: [B, M, d]
        # keys / values z: [B, N, d]
        d = keys.shape(2)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(z) / sqrt(d)
        v = self.v(z)

        # Attention scores [B, M, N]
        attn = q @ k.transpose()

        # Attention probabilities [B, M, N]
        attn = softmax(attn)

        # Value projection [B, M, d]
        out = attn @ v

        return out
```

```
class DenoisingAttention():
    def __init__(self, d):
        # Projections to Q, K, V [d,d]
        self.q = linear(d, d)
        self.k = linear(d, d)
        self.v = linear(d, d)

    def forward(self, u, z, pi):
        # queries       u: [B, M, d]
        # keys / values z: [B, N+1, d]
        d = keys.shape(2)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(z) / sqrt(d)
        v = self.v(z)

        # NVIB bias [B, 1, N+1]
        b = log(pi)
            - 1/(2*sqrt(d))*l2norm(z)**2

        # Attention scores [B, M, N+1]
        attn = q @ k.transpose() + b

        # Attention probabilities [B, M, N+1]
        attn = softmax(attn)

        # Value projection [B, M, d]
        out = attn @ v

        return out
```

Pseudocode: Denoising Attention during evaluation (single-head). Left: Previous implementation including extra bias term and query value interpolation. Right: Current simplified implementation.

```
class DenoisingAttention():
    def __init__(self, d):
        # Projections to Q, K, V [d,d]
        self.q = linear(d, d)
        self.k = linear(d, d)
        self.v = linear(d, d)

    def forward(self, u, mu, sigma2, alpha):
        # queries        u: [B, M, d]
        # keys / values mu: [B, N+1, d]
        d = keys.shape(2)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(mu / (sqrt(d)+sigma2))
        # v is used in interpolation

        # NVIB bias [B, 1, N+1]
        b = log(alpha / sum(alpha))
            - 1/(2*(sqrt(d)+sigma2))*l2norm(mu)**2
            - sum(log(sqrt(sqrt(d)+sigma2)))

        # Attention scores [B, M, N+1]
        attn = q @ k.transpose() + b

        # Attention probabilities [B, M, N+1]
        attn = softmax(attn)

        # Query projection to key-space [B, M, d]
        u_k = self.k(q)

        # Value interpolation [B, M, d]
        out = (attn @ (sigma2/(sqrt(d)+sigma2)))*u_k
             + attn @ ((sqrt(d)/(sqrt(d)+sigma2)))*mu
        out = self.v(out)


        return out
```

```
class DenoisingAttention():
    def __init__(self, d):
        # Projections to Q, K, V [d,d]
        self.q = linear(d, d)
        self.k = linear(d, d)
        self.v = linear(d, d)

    def forward(self, u, mu, alpha):
        # queries      u: [B, M, d]
        # keys/values mu: [B, N+1, d]
        d = keys.shape(2)

        # Project to Q, K, V
        q = self.q(u)
        k = self.k(mu) / sqrt(d)
        v = self.v(mu)

        # NVIB bias [B, 1, N+1]
        b = log(alpha/sum(alpha))
            - 1/(2*sqrt(d))*l2norm(mu)**2

        # Attention scores [B, M, N+1]
        attn = q @ k.transpose() + b

        # Attention probabilities [B, M, N+1]
        attn = softmax(attn)

        # Value projection [B, M, d]
        out = attn @ v

        return out
```

D.4 Attention Plots

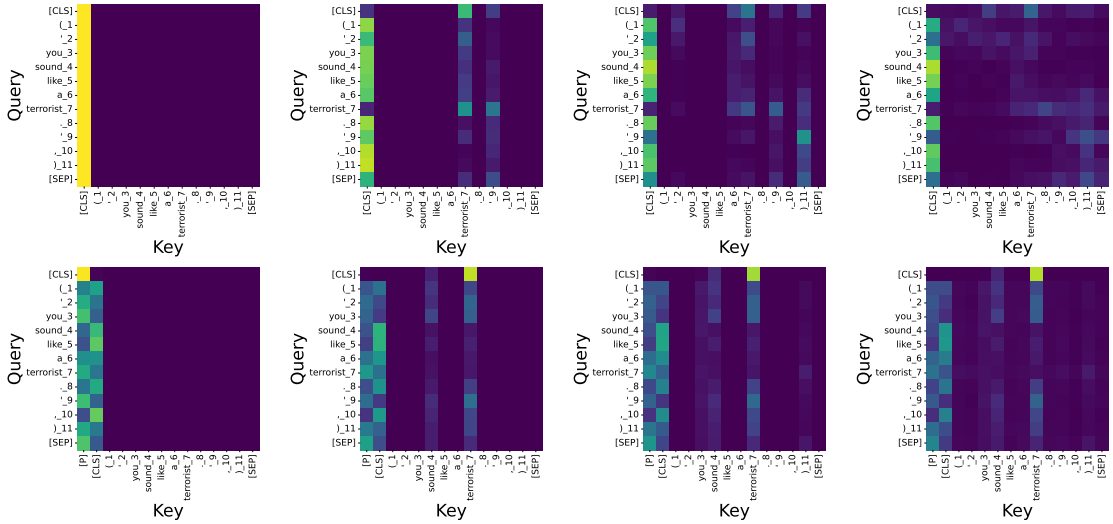


Figure D.1: Attention plot for the best models on CivilComments. The plots show a single head of the last layer. Left-Right: Proportion of keys retained [0.1, 0.25, 0.5, 1.0]. Top: with Dropout. Bottom: with NVIB. Sentence: ('you sound like a terrorist.'). NVIB highlights 'sound' and 'terrorist'.

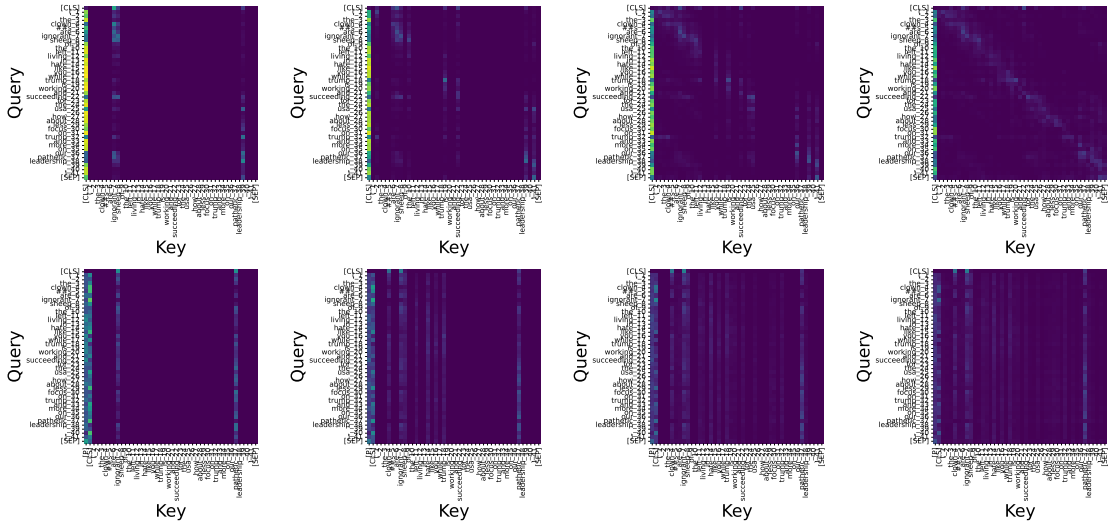


Figure D.2: Attention plot for the best models on CivilComments. The plots show a single head of the last layer. Left-Right: Proportion of keys retained [0.1, 0.25, 0.5, 1.0]. Top: with dropout. Bottom: with NVIB. Sentence: ('the clowns are ignorant sheep of the left living in hate like you while trump is working and succeeding for the usa. how about less focus on trump and more on our pathetic leadership'.) NVIB highlights 'ignorant' and 'pathetic'.

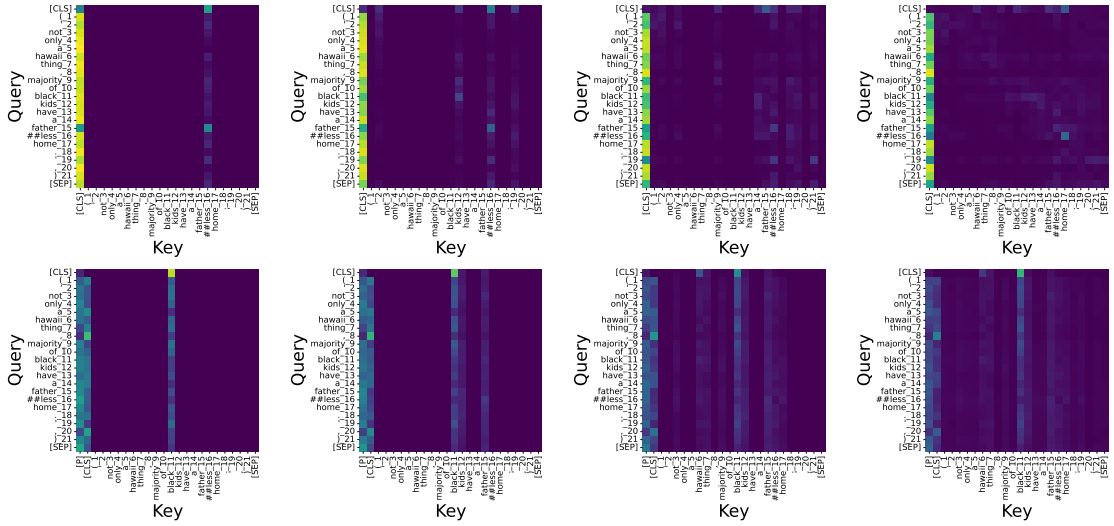


Figure D.3: Attention plots from the best models on CivilComments, showing a single head of the last encoder layer. Left-Right: Proportion of keys retained [0.1, 0.25, 0.5, 1.0]. Top: with Dropout. Bottom: with NVIB. Sentence: ('not only a Hawaii thing, majority of Black kids have a fatherless home.'). NVIB highlights 'Black', 'kids', and 'father'.

Bibliography

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3159–3166, 2019.

David J. Aldous. Exchangeability and related topics. In P. L. Hennequin, editor, *École d'Été de Probabilités de Saint-Flour XIII — 1983*, pages 1–198, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg. ISBN 978-3-540-39316-0.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, Toulon, France, 2017. OpenReview.net. URL https://openreview.net/forum?id=HyxQzBceg.

Duygu Ataman, Wilker Aziz, and Alexandra Birch. A latent morphology model for open-vocabulary neural machine translation. In *International Conference on Learning Representations (ICLR)*, 2020.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, San Diego, CA, USA, May 7-9 2015. URL http://arxiv.org/abs/1409.0473.

Hareesh Bahuleyan, Lili Mou, Olga Vechtomova, and Pascal Poupart. Variational attention for sequence-to-sequence models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1672–1682, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1142.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=p-BhZSz59o4.

Melika Behjati and James Henderson. Inducing meaningful units from character sequences with dynamic capacity slot attention. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=m8U9rSs6gU.

Melika Behjati, Fabio James Fehr, and James Henderson. Learning to abstract with nonparametric variational information bottleneck. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=vU0KbvQ91x.

Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017.

Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxnZh0ct7.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. Generalization in NLI: Ways (not) to go beyond simple heuristics. In João Sedoc, Anna Rogers, Anna Rumshisky, and Shabnam Tafreshi, editors, *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 125–135, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.insights-1.18. URL https://aclanthology.org/2021.insights-1.18.

Srinadh Bhojanapalli, Ayan Chakrabarti, Andreas Veit, Michal Lukasik, Himanshu Jain, Frederick Liu, Yin-Wen Chang, and Sanjiv Kumar. Leveraging redundancy in attention with reuse transformers. *ArXiv*, abs/2110.06821, 2021. URL https://api.semanticscholar.org/CorpusID:238743891.

Yuchen Bian, Jiaji Huang, Xingyu Cai, Jiahong Yuan, and Kenneth Ward Church. On attention redundancy: A comprehensive study. In *North American Chapter of the Association for Computational Linguistics*, 2021. URL https://api.semanticscholar.org/CorpusID:235097467.

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

Gilles Blanchard, Aniket Anand Deshmukh, Ürun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *J. Mach. Learn. Res.*, 22(1), jan 2021. ISSN 1532-4435.

David Blei. Variational inference: Foundations and innovations. *URL http://www. cs. columbia. edu/˜ blei/talks/Blei_VI_tutorial. pdf*, 2017.

David M. Blei and Michael I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121 – 143, 2006. doi: 10.1214/06-BA104. URL https://doi.org/10.1214/06-BA104.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, 2019.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In Stefan Riezler and Yoav Goldberg, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002. URL https://aclanthology.org/K16-1002/.

Mauro Cettolo, Christian Girardi, and Marcello Federico. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy, May 28–30 2012. European Association for Machine Translation. URL https://www.aclweb.org/anthology/2012.eamt-1.60.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan, December 14-15 2017. International Workshop on Spoken Language Translation. URL https://aclanthology.org/2017.iwslt-1.1.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *ArXiv*, abs/1904.10509, 2019. URL https://api.semanticscholar.org/CorpusID:129945531.

Michael Chinen. Marginal effects of language and individual raters on speech quality models. *IEEE Access*, 9:127320–127334, 2021. doi: 10.1109/ACCESS.2021.3112165.

Dokook Choe, Rami Al-Rfou, Mandy Guo, Heeyoung Lee, and Noah Constant. Bridging the gap for tokenizer-free language models. *arXiv preprint arXiv:1908.10322*, 2019.

Christos Christodoulopoulos and Mark Steedman. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49:1–21, 06 2014. doi: 10.1007/s10579-014-9287-y.

Weiqin Chuah, Ruwan Tennakoon, Reza Hoseinnezhad, Alireza Bab-Hadiashar, and David Suter. Itsa: An information-theoretic approach to automatic shortcut avoidance and domain generalization in stereo matching networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13012–13022, 2022. URL https://api.semanticscholar.org/CorpusID:245827816.

Sanghyuk Chun. Improved probabilistic image-text representations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ft1mr3WlGM.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1di0sfgl.

Ondrej Cífka, Aliaksei Severyn, Enrique Alfonseca, and Katja Filippova. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *CoRR*, abs/1804.07972, 2018. URL http://arxiv.org/abs/1804.07972.

Tristan Cinquin, Alexander Immer, Max Horn, and Vincent Fortuin. Pathologies in priors and inference for bayesian transformers. In *Fourth Symposium on Advances in Approximate Bayesian Inference*, 2022. URL https://openreview.net/forum?id=T-3hWOC99vE.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91, 2022. doi: 10.1162/tacl_a_00448. URL https://aclanthology.org/2022.tacl-1.5/.

Alexis Conneau and Douwe Kiela. SentEval: An evaluation toolkit for universal sentence representations. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1269/.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198/.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE, 2023.

Erica Cooper, Wen-Chin Huang, Tomoki Toda, and Junichi Yamagishi. Generalization ability of mos prediction networks. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8442–8446, 2022. doi: 10.1109/ICASSP43922.2022.9746395.

Curation. Curation corpus base, 2020.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Asso-

ciates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/67d57c32e20fd0a7a302cb81d36e40d5-Paper-Conference.pdf.

Daniel Daza, Michael Cochez, and Paul T. Groth. Inductive entity representations from text via link prediction. *Proceedings of the Web Conference 2021*, 2020. URL https://api.semanticscholar.org/CorpusID:222177425.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7480–7512. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/dehghani23a.html.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. Latent alignment and variational attention. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/b691334ccf10d4ab144d672f7783c8a3-Paper.pdf.

Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws. *CoRR*, abs/2212.09720, 2022. URL https://doi.org/10.48550/arXiv.2212.09720.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc., 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain

Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. One size does not fit all: Comparing nmt representations of different granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1504–1516, 2019.

Amine Echraibi, Joachim Flocon-Cholet, Stéphane Gosselin, and Sandrine Vaton. On the variational posterior of dirichlet process deep latent gaussian mixture models. *ArXiv: Computer Science*, abs/2006.08993, 2020. doi: 10.48550/ARXIV.2006.08993. URL https://arxiv.org/abs/2006.08993.

Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. Bayesian attention modules. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16362–16376. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/bcff3f632fd16ff099a49c2f0932b47a-Paper.pdf.

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

William Fedus, Barret Zoph, and Noam M. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23: 120:1–120:39, 2021. URL https://api.semanticscholar.org/CorpusID:231573431.

Fabio James Fehr and James Henderson. Nonparametric variational regularisation of pretrained transformers. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Zu8OWNUC0u.

Martin Ferianc, Ondrej Bohdal, Timothy M. Hospedales, and Miguel L. Rodrigues. Navigating noise: A study of how noise influences generalisation and calibration of neural networks. *Transactions of Machine Learning Research*, 2024, 2023.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/92c8c96e4c37100777c7190b76d28233-Paper.pdf.

Elias Frantar, Sidak Pal Singh, and Dan Alistarh. Optimal Brain Compression: a framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 36, 2022.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=tcbBPnfwxS.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=tcbBPnfwxS.

Bela A Frigyik, Amol Kapila, and Maya R Gupta. Introduction to the dirichlet distribution and related processes. *Department of Electrical Engineering, University of Washignton, UWEETR-2010-0006*, 6:1–27, 2010.

R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2 (11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. From variational to deterministic autoencoders. In *8th International Conference on Learning Representations, ICLR 2020, Conference Track Proceedings*, 2020. URL https://openreview.net/forum?id=S1g7tpEYDS.

H.W. Gierlich and F. Kettler. Advanced speech quality testing of modern telecommunication equipment: An overview. *Signal Processing*, 86(6):1327–1340, 2006. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2005.06.024. URL https://www.sciencedirect.com/science/article/pii/S0165168405003312. Applied Speech and Audio Processing.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL https://aclanthology.org/D19-5409.

Prasoon Goyal, Zhiting Hu, Xiaodan Liang, Chenyu Wang, Eric P. Xing, and Carnegie Mellon. Nonparametric variational auto-encoders for hierarchical representation learning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5104–5112, 2017. URL https://openaccess.thecvf.com/content_ICCV_2017/papers/Goyal_Nonparametric_Variational_Auto-Encoders_ICCV_2017_paper.pdf.

Zongyan Han, Zhenyong Fu, and Jian Yang. Learning the redundancy-free features for generalized zero-shot object recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12862–12871, 2020. URL https://api.semanticscholar.org/CorpusID:219633468.

James Henderson. The unstoppable rise of computational linguistics in deep learning. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6294–6306, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.561. URL https://aclanthology.org/2020.acl-main.561.

James Henderson and Fabio James Fehr. A VAE for transformers with nonparametric variational information bottleneck. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6QkjC_cs03X.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. Conditional probing: measuring usable information beyond a baseline. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.122. URL https://aclanthology.org/2021.emnlp-main.122/.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.43. URL https://aclanthology.org/2022.acl-short.43/.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Shell Xu Hu, Da Li, Jan Stühmer, Minyoung Kim, and Timothy M. Hospedales. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference. In *CVPR*, 2022b.

Itay Hubara, Brian Chmiel, Moshe Island, Ron Banner, Joseph Naor, and Daniel Soudry. Accelerated sparse neural training: A provable and efficient method to find n:m transposable masks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21099–21111. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b0490b85e92b64dbb5db76bf8fca6a82-Paper.pdf.

Maximilian Ilse, Jakub M. Tomczak, Christos Louizos, and Max Welling. DIVA: Domain invariant variational autoencoder. In *Medical Imaging with Deep Learning*, 2020. URL https://openreview.net/forum?id=RmNckVums7.

ITU-T. Methods for subjective determination of transmission quality. Recommendation P.800, International Telecommunication Union, Geneva, Switzerland, August 1996.

ITU-T. Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models. Recommendation P.1401, International Telecommunication Union, Geneva, Switzerland, January 2020.

M.I. Jordan. Bayesian nonparametric learning: Expressive priors for intelligent systems. In R. Dechter, H. Geffner, and J. Halpern, editors, *Heuristics, Probability and Causality: A Tribute to Judea Pearl*, chapter 10. College Publications, 2010.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P18-4020.

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. How does information bottleneck help deep learning? In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16049–16096. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kawaguchi23a.html.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. Learning to create and reuse words in open-vocabulary neural language modeling. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1492–1502, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1137. URL https://aclanthology.org/P17-1137/.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings*, Banff, AB, Canada, 2014. URL http://arxiv.org/abs/1312.6114.

Diederik P. Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational autoencoders with inverse autoregressive flow. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 4736–4744, Barcelona, Spain, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/ddeebdeefdb7e7e7a697e1c3e3d8ef54-Abstract.html.

David A. Knowles. Stochastic gradient variational bayes for gamma approximating distributions. *arXiv: Machine Learning*, 2015. URL https://arxiv.org/abs/1509.01631.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts, 2021.

Jannik Kossen, Neil Band, Clare Lyle, Aidan N Gomez, Thomas Rainforth, and Yarin Gal. Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 28742–28756. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/f1507aba9fc82ffa7cc7373c58f8a613-Paper.pdf.

Mahnaz Koupaee and William Yang Wang. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305, 2018. URL http://arxiv.org/abs/1810.09305.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012/.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2020.124289. URL https://www.sciencedirect.com/science/article/pii/S0378437120300856.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703/.

Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Colorado Reed, Jun Zhang, Dongsheng Li, Kurt Keutzer, and Han Zhao. Invariant information bottleneck for domain generalization. In *AAAI Conference on Artificial Intelligence*, 2021a. URL https://api.semanticscholar.org/CorpusID:235417355.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5543–5551, 2017. URL https://api.semanticscholar.org/CorpusID:6037691.

Pengyong Li, Jun Wang, Yixuan Qiao, Hao Chen, Yihuan Yu, Xiaojun Yao, Peng Gao, Guowang Xie, and Sen Song. An effective self-supervised framework for learning expressive molecular global representations to drug discovery. *Briefings in bioinformatics*, 2021b. URL https://api.semanticscholar.org/CorpusID:233719686.

Jiachen Lian, Chunlei Zhang, and Dong Yu. Robust disentangled variational speech representation learning for zero-shot voice conversion. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6572–6576, 2022. URL https://api.semanticscholar.org/CorpusID:247839160.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Danyang Liu and Gongshen Liu. A transformer-based variational autoencoder for sentence generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019. doi: 10.1109/IJCNN.2019.8852155. URL https://arxiv.org/abs/2003.12738.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=rkgz2aEKDr.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=rkgz2aEKDr.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Rabeeh Karimi mahabadi, Yonatan Belinkov, and James Henderson. Variational information bottleneck for effective low-resource fine-tuning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=kvhzKz-_DMF.

Alice Martin, Charles Ollion, Florian Strub, Sylvain Le Corff, and Olivier Pietquin. The monte carlo transformer: a stochastic self-attention model for sequence prediction. *ArXiv: Mathematics, Computer Science*, abs/2007.08620, 2020. URL https://arxiv.org/abs/2007.08620.

Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4402–4412, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL https://arxiv.org/abs/1812.02833.

Arya D. McCarthy, Xian Li, Jiatao Gu, and Ning Dong. Addressing posterior collapse with mutual information for improved variational neural machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://api.semanticscholar.org/CorpusID:220046608.

Giangiacomo Mercatali and André Freitas. Disentangling generative factors in natural language with discrete variational autoencoders. In *Findings of the Association for*

Computational Linguistics: EMNLP 2021, pages 3547–3556, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.301. URL https://aclanthology.org/2021.findings-emnlp.301.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, Toulon, France, 2017. OpenReview.net. URL https://openreview.net/forum?id=Byj72udxe.

Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *Interspeech 2021*, Aug 2021. doi: 10.21437/interspeech.2021-299. URL http://dx.doi.org/10.21437/Interspeech.2021-299.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, 2020.

K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10 (1-2):1–141, 2017. doi: 10.1561/2200000060.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, Haifa, Israel, 21–24 June 2010. Omnipress. URL https://icml.cc/Conferences/2010/papers/432.pdf.

Eric T. Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, Toulon, France, April 24-26 2017. OpenReview.net. URL https://openreview.net/forum?id=S1jmAotxg.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206.

Alexandru Nelus and Rainer Martin. Privacy-preserving audio classification using variational information feature extraction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2864–2877, 2021. URL https://api.semanticscholar.org/CorpusID:237518792.

Tam Minh Nguyen, Tan Minh Nguyen, Dung D. D. Le, Duy Khuong Nguyen, Viet-Anh Tran, Richard Baraniuk, Nhat Ho, and Stanley Osher. Improving transformers with probabilistic attention keys. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16595–16621. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nguyen22c.html.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. An information bottleneck approach for controlling conciseness in rationale extraction. In *Conference on Empirical Methods in Natural Language Processing*, 2020. URL https://api.semanticscholar.org/CorpusID:218487373.

Seongmin Park and Jihwa Lee. Finetuning pretrained transformers into variational autoencoders. In João Sedoc, Anna Rogers, Anna Rumshisky, and Shabnam Tafreshi, editors, *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 29–35, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.insights-1.5. URL https://aclanthology.org/2021.insights-1.5/.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162/.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202/.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October

2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140): 1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. Speechbrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*, 2021.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying WEI, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12559–12571. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/94aef38441efa3380a3bed3faf1f9d5d-Paper.pdf.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:186206211.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), mar 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and Philip S. Yu. Graph structure learning with variational information bottleneck. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):4165–4174, 2022. ISSN 2374-3468. doi: 10.1609/aaai.v36i4.20335. 36th AAAI Conference on Artificial Intelligence, AAAI 2022 ; Conference date: 22-02-2022 Through 01-03-2022.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JtBRnrlOEFN.

Yee Whye Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*, volume 1063, pages 280–287. Springer, 2010.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015. URL https://api.semanticscholar.org/CorpusID:5541663.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *ArXiv*, physics/0004057, 2000. URL https://api.semanticscholar.org/CorpusID:8936496.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers and distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. URL https://api.semanticscholar.org/CorpusID:257219404.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2019.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv: Computation and Language*, 2019. URL https://api.semanticscholar.org/CorpusID:202889175.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e3b21256183cf7c2c7a66be163579d37-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.80. URL https://aclanthology.org/2021.acl-long.80.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Jiawei Wu, Xiaoya Li, Xiang Ao, Yuxian Meng, Fei Wu, and Jiwei Li. Improving robustness and generality of nlp models using disentangled representations. *ArXiv*, abs/2009.09587, 2020a. URL https://api.semanticscholar.org/CorpusID:221819589.

Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20437–20448. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ebc2aa04e75e3caabda543a1317160c0-Paper.pdf.

Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2:109–127, 2021. URL https://api.semanticscholar.org/CorpusID:233481068.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10: 291–306, 2022. doi: 10.1162/tacl_a_00461. URL https://aclanthology.org/2022.tacl-1.17/.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27168–27183. Curran Associates, Inc., 2022.

Gaoxiong Yi, Wei Xiao, Yiming Xiao, Babak Naderi, Sebastian Möller, Wafaa Wardah, Gabriel Mittag, Ross Culter, Zhuohuang Zhang, Donald S. Williamson, Fei Chen, Fuzheng Yang, and Shidong Shang. ConferencingSpeech 2022 Challenge: Non-intrusive Objective Speech Quality Assessment (NISQA) Challenge for Online Conferencing Applications. In *Proc. Interspeech 2022*, pages 3308–3312, 2022. doi: 10.21437/Interspeech.2022-10597.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online, July 2020.

Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.148. URL https://aclanthology.org/2020.acl-main.148.

Cenyuan Zhang, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. Improving the adversarial robustness of NLP models by information bottleneck. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3588–3598, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.284. URL https://aclanthology.org/2022.findings-acl.284/.

Min Zhang, Haoxuan Li, Fei Wu, and Kun Kuang. Metacoco: A new few-shot classification benchmark with spurious correlation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DiWRG9JTWZ.

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

Chenye Zhao, Jasmine Mangat, Sujay Koujalgi, Anna Squicciarini, and Cornelia Caragea. Privacyalert: A dataset for image privacy prediction. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):1352–1361, May 2022. doi: 10.1609/icwsm.v16i1.19387. URL https://ojs.aaai.org/index.php/ICWSM/article/view/19387.

Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. Adversarially regularized autoencoders. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5902–5911. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/zhao18b.html.

Guangtao Zheng, Wenqian Ye, and Aidong Zhang. Benchmarking spurious bias in few-shot image classifiers. In *European Conference on Computer Vision*, 2024. URL https://api.semanticscholar.org/CorpusID:272398061.

Fabio J. Fehr

PhD at EPFL & Idiap

25/05/1996,
Swiss & South African

fabio.fehr@idiap.ch

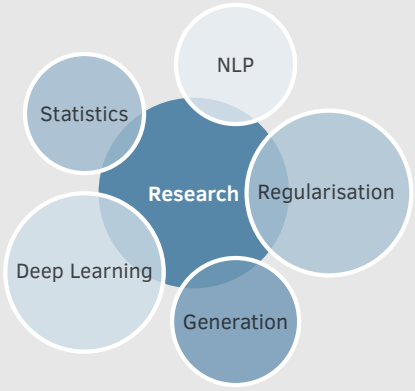
LinkedIn Profile

Research Profile

About me

I advocate for leadership by example. I believe that the values: honesty, humility and kindness are the ultimate form of optimisation – free to give and invaluable to receive. I am passionate and tenacious in my research and think that the progress direction is more important than the progress magnitude. My happiest state: rigorous mental stimulation, relaxing social interaction and regular outdoor exercise. For leisure, I enjoy spending time in nature: sunshine, sea and snowy mountains.

Interests



Languages

- English
- French
- Afrikaans
- German

Education

Tertiary: Postgraduate

2021 – 2025 **Doctor of Philosophy: Electrical Engineering** EPFL
Research in machine learning and natural language processing.

PhD Research EPFL
Combining Variational Bayesian nonparametric methods with deep attention-based models for representation learning.

2019 – 2020 **Masters: Advanced Analytics** UCT
Statistics specialisation and awarded SASA-NRF full bursary.

Masters Thesis UCT
A comprehensive comparison study of Gaussian processes and variational autoencoders for statistical shape modelling of 3D meshes.

Tertiary: Undergraduate

2015 – 2018 **Bachelors of Business Science: Analytics** UCT
Commerce degree with a specialisation in mathematics & statistics.

Honours Thesis (Distinction) UCT
Statistical machine learning techniques for classifying news articles. Presented at South African Statistical Association (SASA) 2019.

Experience

2024 **Intern Applied Data Scientist** Amazon
Retrieval and augmented generation (RAG) with large language models (LLMs) for code generation.

2021-2025 **Research Assistant** Idiap
Natural Language Processing and Deep Learning research.

2018 **Intern Machine Learning Engineer** DataProphet
Machine learning for financial time-series data.

2017 - 2018 **Intern Data Scientist** Eighty20
Large-scale data processing and visualisation.

Teaching

2021-2024 **Teaching Assistant** EPFL
Machine learning, Deep learning and NLP courses at EPFL.

2019-2024 **Subject Matter Expert & Engagement Tutor** 2U
Business Analytics (UCT), Machine Learning (LSE).

2017 – 2020 **Head Tutor: Statistics** UCT
Managing, coordination, tutoring and assistant lecturing.

Publications

PhD Research

2025 Fine-Tuning Pretrained Models with NVIB for Improved Generalisation ICLR 2025

2024 Nonparametric Variational Regularisation of Pretrained Transformers COLM 2024

2023 Learning to Abstract with Nonparametric Variational Information Bottleneck EMNLP 2023

2022 A VAE for Transformers with Nonparametric Variational Information Bottleneck ICLR 2023

Research Collaborations

2023 HyperMixer: An MLP-based Low Cost Alternative to Transformers ACL 2023

2025 Coret: Improved Retriever for Code Editing ACL 2025

References

PhD supervisor Idiap
Dr. James Henderson - james.henderson@idiap.ch

Internship Mentor Amazon
Dr. Prabhu Teja Sivaprasad - prbuteja@amazon.de