Learning to Disentangle Latent Reasoning Rules with Language VAEs: A Systematic Study

Yingji Zhang[1,3], **Marco Valentino**[2], **Danilo S. Carvalho**[1,4], **André Freitas**[1,3,4]
[1]Department of Computer Science, University of Manchester, UK
[2]School of Computer Science, University of Sheffield, UK
[3]Idiap Research Institute, Switzerland
[4]CRUK National Biomarker Centre, University of Manchester, UK
[1]{firstname.lastname}@manchester.ac.uk, [2]{firstname.lastname}@sheffield.ac.uk

Abstract

Incorporating explicit reasoning rules within the latent space of language models (LMs) offers a promising pathway to enhance generalisation, interpretability, and controllability. While current Transformer-based language models have shown strong performance on Natural Language Inference (NLI) tasks, they often rely on memorisation rather than explicit rule-based generalisation. This work investigates how human-interpretable reasoning rules can be explicitly encoded within LMs with the support of Language Variational Autoencoders (VAEs), as a mechanism for generative control. We propose a complete pipeline for learning reasoning rules within Transformer-based language VAEs. This pipeline encompasses three rule-based reasoning tasks, a supporting theoretical framework, and a practical end-to-end architecture. The experiment illustrates the following findings: **Disentangled reasoning:** Under explicit signal supervision, reasoning rules (viewed as functional mappings) can be disentangled within the encoder's parametric space. This separation results in distinct clustering of rules in the output feature space. **Prior knowledge injection:** injecting rule-based constraints into the Query enables the model to more effectively retrieve the stored Value from memory based on Key. This approach offers a simple method for integrating prior knowledge into decoder-only language models. Moreover, we found that FFN layers are better than attention layers at preserving the separation of reasoning rules in the model's parameters.

Introduction

Encoding reasoning patterns as explicit rules within latent representations holds significant promise for enhancing the generalisation, interpretability, and controllability of neural models, with high downstream impact on AI safety and regulatory compliance (Bonnet and Macfarlane 2024; Yu, Chatzi, and Kissas 2025). Over recent years, Transformer-based language models (LMs) have achieved notable success across a variety of Natural Language Inference (NLI) tasks (Qwen et al. 2025). Nonetheless, a growing body of research has demonstrated that in many instances these models often rely on memorisation rather than generalisation and that rule-based control mechanisms fail to be fully enforced (Yan et al. 2025).

Therefore, this investigation focuses on the question: *How can reasoning rules be explicitly encoded within the latent space of language models?* In the context of this work, a reasoning rule is defined as an input–output pattern that reflects a specific transformation of inference behaviour. It is important to note that this study does not aim to explore rule composition or generalisation. Rather, the primary objective is to explicitly encode reasoning input–output constraints within the model's latent space, targeting better interpretability and controllability in the latent space.

Variational Autoencoders (VAEs) (Kingma and Welling 2013) provide a compelling framework for this direction, where the integration of prior distribution serves as an inductive bias, enabling the model to leverage existing knowledge and providing a principled way to incorporate domain constraints (Papamarkou et al. 2024). This work focuses on lightweight LMs ($< 1B$) as VAE decoders, allowing accessible evaluation of training dynamics, memory capacity, and model updates of the Transformer architecture (Zhong et al. 2025; Morris et al. 2025). Additionally, because our primary motivation is to study latent rule learning within the encoder latent space, we fully re-train the decoder. As a result, the geometrical properties of the latent space are not constrained by the model's architecture, enabling a more controlled analysis of representation learning.

Accordingly, we propose a complete pipeline for learning NLI reasoning rules within Transformer-based language VAEs (Li et al. 2020; Zhang et al. 2024b):

First, our study centers on three rule-driven reasoning tasks, which differ in their syntactic structures, inference mechanisms, and granularity, including (i) *Mathematical Reasoning* (Meadows et al. 2023), (ii) *Syllogistic Reasoning* (Valentino et al. 2025), and (iii) *Explanatory Reasoning* (Zhang et al. 2024a).

Second, following the Neural Tangent Kernel (NTK) theory (Jacot, Gabriel, and Hongler 2018), we posit that supervising inference-rule information enables these rules to be encoded in the encoder's parameter space (i.e., its weight matrices), which in turn induces separation of rules in the latent sentence space. This perspective provides a conceptual framework for explicitly embedding reasoning rules into the latent representation space.

Third, we propose an end-to-end VAE architecture where the latent space can encode both coarse-grained reasoning

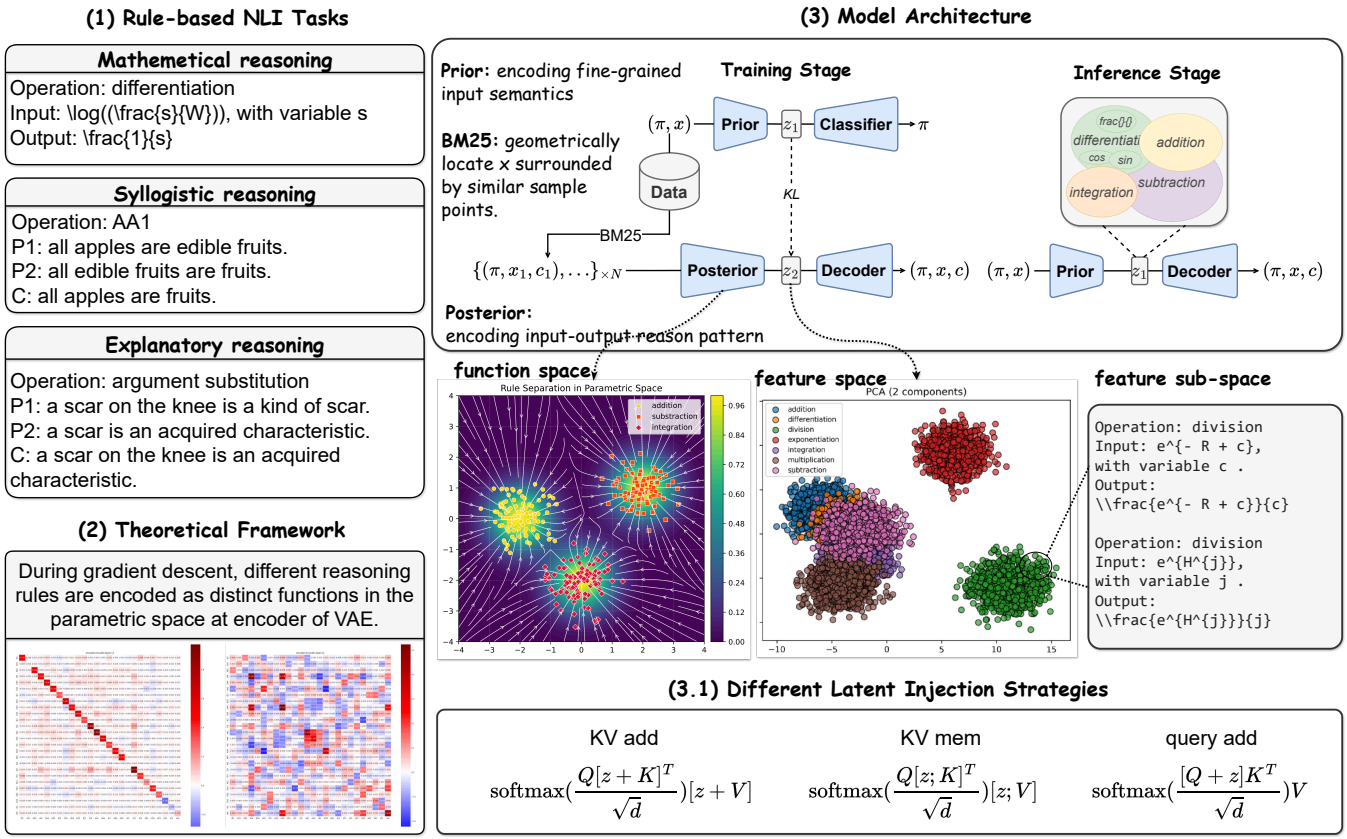

Figure 1: Overview, where (π, x, c) represents the *(rule, input premise(s), conclusion)*. To systematically evaluate rule-based learning within a VAE framework, first, we examine three rule-based NLI tasks. Second, we formalise the hypothesis that reasoning rules can be functionally and separately encoded within the encoder's parametric space, enabling rule learning in the latent space, grounded in the theoretical framework of neural tangent kernels. Third, we introduce an end-to-end VAE architecture, with three different latent injection setups, designed to capture coarse-grained reasoning patterns in its latent space while remaining sensitive to the lexical semantics of the input.

rules and fine-grained reasoning patterns that arise from semantic differences in the inputs, and evaluate three ways to inject the latent space into the LM decoder by intervening in the attention network.

We conduct extensive experiments to assess the effectiveness of rule encoding and reasoning generation capabilities which elicits the following results and findings:

Disentangled Reasoning: Under explicit signal supervision, reasoning rules, as functional mappings, can be disentangled within the encoder's parametric space. This separation results in distinct clustering of rules in the output feature space, suggesting the potential for employing rule-based NTK theory to better understand the training dynamics and internal geometry of gradient-based neural NLI model.

Prior Knowledge Injection: Different latent space injection setups result in varying levels of reasoning performance. The optimal configuration is achieved by injecting the latent space into the Query of the attention network. Intuitively, injecting rule-based reasoning constraints into the Query enables the model to more effectively retrieve the stored Value from memory based on Key. This approach offers a simple method for integrating prior knowledge into LMs.

Information Bottleneck: A case study on the mathematical reasoning task reveals a performance bottleneck in decoder-only LMs, where increasing the number of samples per operation fails to yield performance improvements beyond a certain threshold. Additionally, we observe that FFN networks are more effective than attention in maintaining the separation of reasoning rules within the learned parametric space. Both provide additional insights on understanding Transformer architectures through memorisation.

To the best of our knowledge, this is the first study to explore the explicit encoding of NLI rules within language VAE latent spaces, targeting better latent space geometry to support rule-based inferences.

Related Work

In this section, we review related work around two topics: *rule-based representation learning* and *language VAEs*, to highlight current research limitations and elucidate the motivation underlying our work.

Rule-based Representation Learning. Encoder-only models have been employed for tasks such as mathematical operations, where the encoder learns structured transformations, as demonstrated by (Valentino et al. 2024). In addition, VAE-based approaches have shown promise in tasks requiring structured reasoning, such as program synthesis, where the goal is to generate programs that fulfil a specified task (Sun et al. 2018; Bonnet and Macfarlane 2024; van Krieken et al. 2025). Similarly, grammar-based approaches using VAEs have been applied to infer ordinary differential equation (ODE) formulas from data (Yu, Chatzi, and Kissas 2025). Moreover, Decoder-only models, particularly large language models (LLMs), leverage in-context learning by using demonstrations to infer and apply underlying reasoning patterns (Liu, Xing, and Zou 2023; Bhattamishra et al. 2024). Despite these advancements, relatively few studies have investigated rule-based learning in the context of NLI, where this study focuses on.

Language VAEs. Language VAEs have been widely applied in NLP tasks, such as style transfer tasks: modifying sentences with regard to markers of sentiment, formality, affirmation/negation (Shen et al. 2020; John et al. 2019; Bao et al. 2019a; Hu and Li 2021; Vasilakes et al. 2022; Gu et al. 2022; Liu et al. 2023; Gu et al. 2023), story generation (Fang et al. 2021), dialogue generation (Zhao, Zhao, and Eskenazi 2017), text paraphrasing (Bao et al. 2019b), and textual, syntactic, semantic representation learning domain, such as syntax disentanglement (Mercatali and Freitas 2021), semantic-syntax separation (Zhang et al. 2024b), semantic disentanglement (Carvalho et al. 2023; Zhang, Carvalho, and Freitas 2024), etc. Comparatively, in this work we focus on Natural Language Inference (NLI) with an emphasis on rule-based control.

In the next section, we start by introducing the reasoning tasks and provide a formal illustration of rule-based learning through Neural Tangent Kernel theory.

Rule-based Learning for NLI

Natural Language Inference Rules

We investigate three distinct types of reasoning tasks: mathematical derivation (Meadows et al. 2024), syllogistic reasoning (Valentino et al. 2025), and explanatory reasoning (Zhang et al. 2024a). Specifically, **(1)** Mathematical reasoning: Mathematical expressions (Valentino et al. 2023; Meadows et al. 2023) follow a well-defined syntactic structure and set of symbolic rules that are notoriously difficult for neural models. The dataset (Meadows et al. 2023) includes seven human-annotated symbolic rules, encompassing operations such as *differentiation* and *integration*. **(2)** Syllogistic reasoning: Syllogistic reasoning involves classical categorical logic, including four standard forms: Universal Affirmative (A) - "All A are B", Universal Negative (E) - "No A are B", Particular Affirmative (I) - "Some A are B", and Particular Negative (O) - "Some A are not B". The dataset encodes 24 valid syllogistic inference patterns, such as AA1 (Barbara). **(3)** Explanatory reasoning: Consists of material inferences with clearly defined explanatory sentence structures (Jansen et al. 2018; Valentino et al. 2022) providing a semantically challenging yet sufficiently well-scoped scenario to evaluate the syntactic and semantic organisation of the space. Based on the EntailmentBank corpus (Dalvi et al. 2021), Zhang et al. (2024a) we can define ten inference types based on these explanatory patterns, providing a diverse set of tasks which instantiate rule-based reasoning. A summarisation of each task and corpus is provided in Table 4 in the supplementary material.

Rule-based Neural Tangent Kernel

This work aims to encode latent reasoning rules within a low-dimensional latent space to guide the LM decoder's reasoning generation, leveraging the VAE framework. We frame rule-based learning as learning the transformation from input premise(s) to output conclusion within the encoder. We formalise this framework by introducing a novel method grounded in the *Neural Tangent Kernel (NTK) theory* (Jacot, Gabriel, and Hongler 2018).

Latent subspace separation. Let $\mathcal{M}$ be a VAE model parameterised by $\theta = (\theta_{\text{enc}}, \theta_{\text{dec}})$. Suppose the encoder can represent a set of symbolic inference rules $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$. Then, under supervised training with rule annotations:

Proposition: *the encoder's parameters θ_{enc} induce a parametric structure in which each inference rule π_i corresponds to a distinct subspace $S_{\pi_i} \subseteq \mathbb{R}^D$ of the encoder representation space.*

$$\forall \pi_i, \pi_j \in \Pi,\ \pi_i \neq \pi_j \quad \Rightarrow \quad S_{\pi_i} \cap S_{\pi_j} \approx \emptyset \tag{1}$$

This parametric separation (Figure 2) directly leads to clearly delineated latent feature subspaces (Figure 3), each uniquely encoding a symbolic inference rule.

Connection to NTK theory. NTK theory provides a rigorous theoretical framework for understanding neural network training dynamics by examining the kernel induced by gradients of the network's outputs with respect to its parameters. It suggests that during gradient descent, the network effectively learns linear approximations within distinct functional subspaces, each approximating discrete symbolic reasoning rules within the target reasoning task. Formally, let f_{encode} be the encoder function such that:

$$f_{\text{encode}} : (\pi, x, c) \mapsto \mathbf{z}_{(\pi,x,c)} \in \mathcal{Z} \subseteq \mathbb{R}^D \tag{2}$$

where x is the input premise(s), c is the conclusion, $\pi \in \Pi$ is the reasoning rule, and $\mathcal{Z}$ is the latent representation space of dimension D. Each rule π is explicitly embedded as part of the model input. As a result, the model effectively learns a function $f_\theta(\pi, x, c)$. The function f_θ thus jointly depends on both the content of transformation from premise(s) to conclusion and the nature of the symbolic operation to be performed. Within the NTK framework, the similarity between two input examples of the same inference type π is captured by the NTK as follows: $\Theta_\pi(x, x') = \nabla_\theta f_\theta(x, \pi)^\top \nabla_\theta f_\theta(x', \pi)$, where x represents (x, c) pair for concision. $\nabla_\theta f_\theta(x, \pi)$ denotes the gradient of the model output with respect to its parameters, evaluated at the input (x, π). This kernel quantifies how a parameter update from

one input-output pair would affect another pair, conditioned on the shared rule.

According to the NTK theory, the evolution of the model's predictions under gradient descent training can be described by a linear kernel regression in the Reproducing Kernel Hilbert Space (RKHS) associated with Θ_π. Crucially, this formulation implies that each rule π induces a distinct kernel Θ_π, which in turn defines a unique RKHS, that is, a function space within which the model's solutions for rule π reside. As the π is varied, the structure of the kernel and the corresponding function space changes, reflecting the distinct reasoning behaviours associated with different inference operations. Thus, the model encodes different symbolic inference patterns in distinct, kernel-induced subspaces.

Given two different reasoning rules, $\pi_i \neq \pi_j$, we examine the relationship between their corresponding NTKs, Θ_{π_i} and Θ_{π_j}. Specifically, we examine the interactions between parameter gradients that are induced by inputs corresponding to different types of inference. Given two data points x and x', possibly corresponding to different premise pairs, the NTK entry for each rule is: $\Theta_{\pi_i}(x, x') = \nabla_\theta f_\theta(x, \pi_i)^\top \nabla_\theta f_\theta(x', \pi_i)$ and $\Theta_{\pi_j}(x, x') = \nabla_\theta f_\theta(x, \pi_j)^\top \nabla_\theta f_\theta(x', \pi_j)$ When considering cross-rule similarities, we are interested in the inner product between the gradients for different rules. If the rules π_i and π_j encode different reasoning operations (e.g., *addition* vs. *subtraction* in mathematical derivation), then the gradients with respect to θ for inputs labeled with π_i and π_j will generally point in different directions within the encoder's parameter space.

Under idealised training, where the data for each rule is sufficiently distinct and the network has enough capacity, the gradients for one rule will have minimal overlap with those of the other. This can be formalised by observing that:

$$\langle \nabla_\theta f_\theta(x, \pi_i), \nabla_\theta f_\theta(x', \pi_j) \rangle \approx 0 \qquad \text{for } \pi_i \neq \pi_j \tag{3}$$

This property implies that the parameter updates driven by examples from different rules are approximately orthogonal, meaning that training on one type will not interfere with or alter the function learned for the other type. In the language of NTK and kernel regression, this corresponds to the induced RKHS for each type, $\mathcal{H}\pi_i$ and $\mathcal{H}\pi_j$, being approximately disjoint:

$$\mathcal{H}\pi_i \cap \mathcal{H}\pi_j \approx \emptyset \tag{4}$$

In the next section, we will use this foundation to introduce the proposed VAE architecture and its supporting optimisation function.

Approach

Latent space properties. We posit that the latent space should satisfy two essential geometrical properties:

Property 1: Rule-level encoding. The latent space must capture the transformation defined by the reasoning rule π, which maps an input x to a conclusion c, i.e., $\pi : x \to c$.

Property 2: Semantic-level encoding. The latent space should also encode the lexical semantics of x, accounting for fine-grained variations in reasoning patterns that arise from semantic differences in the inputs. For example, within the Math Derivation task, under *differentiation*, different inputs yield distinct transformations: $\pi : x_1 \to c_1$: 8 \sin{(u)} $\to$ 8 \cos{u} and $\pi : x_2 \to c_2$: \log{K} $\to$ \frac{1}{K}. By ensuring both properties, the latent space can capture both coarse-grained and fine-grained reasoning patterns.

Architecture. We adopt a Transformer-based VAE framework, employing two distinct encoders, both instantiated with BERT (Devlin et al. 2018), to model the prior and posterior Gaussian distributions. The posterior encoder is optimised to capture the reasoning rule transformation, thereby satisfying *Property 1*. Simultaneously, the prior encoder serves as a regulariser, encouraging the posterior to encode fine-grained sub-rule variations grounded in the lexical semantics of the input, thereby satisfying *Property 2*.

Concretely, given a target (π, x) pair, we first retrieve a set of semantic similar examples $\{(\pi, x_1, c_1), \ldots, (\pi, x_N, c_N)\}$ using a retrieval function (e.g. BM25 and N=12). These examples are defined as inputs to the posterior encoder, which learns a latent representation of the reasoning transformation by averaging the latent vectors of the retrieved instances. Geometrically, averaging the latent sample vectors positions the target x at the centroid of the surrounding samples within the latent space, which is naturally aligned with how information is encoded in the latent space (Zhang, Carvalho, and Freitas 2025). The decoder then uses this aggregated representation to generate the corresponding conclusion c for the target triplet (π, x, c). In parallel, the prior encoder processes the target (π, x) pair to predict the rule π within the latent space via a linear classifier. By minimising the Kullback–Leibler (KL) divergence between the posterior and prior distributions, the latent space is encouraged to satisfy both geometric properties.

Latent injection. We adopt three strategies to inject the latent variable z into the decoder (e.g., Qwen2.5 (Qwen et al. 2025)). Each approach modifies the attention mechanism to incorporate information from the latent space.

1. `kv_add` Following (Zhang et al. 2024b), the latent vector z is added to both the Key (K) and Value (V) matrices in the attention network: softmax $\left(\frac{\mathrm{Q[z+K]}^\top}{\sqrt{d}}\right)$ [z + V]

2. `kv_mem` Following (Li et al. 2020), the latent vector z is concatenated to the Key and Value matrices: softmax $\left(\frac{\mathrm{Q[z;K]}^\top}{\sqrt{d}}\right)$ [z; V]

3. `query_add` In this setup, the latent vector z is added to the Query (Q) matrix: softmax $\left(\frac{\mathrm{[Q+z]K}^\top}{\sqrt{d}}\right)$ V

Here, Q, K, and V denote the query, key, and value matrices in the attention mechanism, each with dimensions $\mathbb{R}^{\mathrm{dim} \times \mathrm{seq}}$, where dim is the attention dimensionality and seq is the sequence length.

Optimisation. Finally, the model can be trained end-to-end via the evidence lower bound (ELBO) on the log-likelihood of the data x (Kingma and Welling 2013). To avoid the KL vanishing issue, we select the cyclical schedule to increase weights of KL β from 0 to 1 (Fu et al. 2019) and a KL thresholding scheme (Li et al. 2019) that chooses the

Base Model	Math Derivations		Math Derivations (OOD)		Syllogistic Reasoning		Explanatory Reasoning
	bleu	acc	bleu	acc	bleu	acc	bleu
GPT2-medium	0.2019	0.0171	0.0206	0.0018	0.0108	0.0000	0.5947
Llama3-1B	0.3412	0.0257	0.0625	0.0114	0.3612	0.1200	0.3160
Qwen2-0.5B	0.4200	0.1171	0.0869	0.0128	0.8130	0.4600	0.5372
Qwen2.5-0.5B	**0.6260**	**0.3800**	**0.1293**	**0.0185**	**0.9014**	**0.7000**	**0.6566**

Table 1: Quantitative evaluation for decoder-only models. We can observe that Qwen2.5-0.5B demonstrates strong performance across various baselines, making it a suitable choice as the decoder component in our VAE architecture.

maximum between KL and threshold λ. The final objective function can be described as follows:

$$\begin{aligned}\mathcal{L}_{\text{VAE}} =& \mathbb{E}_{q_\phi(z|\pi,x,c)}\Big[\log p_\theta(\pi,x,c|z)\Big] \\ &- \beta \max\left[\lambda, \text{KL}q_\phi(z|\pi,x,c)||p(z|\pi,x)\right] \\ &+ \texttt{cls\_weight} \times \mathcal{L}_{\text{classifier}}\end{aligned} \tag{5}$$

where q_ϕ, p, and p_θ represent the posterior encoder, prior encoder, and decoder, respectively. `cls_weight` controls the strength of the classification loss. In our experiments, `cls_weight` is evaluated at 1.0, 0.5, and 0.1.

Empirical Analysis

Decoding Evaluation

Base model. First, we evaluate the reasoning performance of different decoder-only models, including Qwen2 and Qwen2.5 (Qwen et al. 2025), GPT2 (Radford et al. 2019), and trimmed Llama3. Due to the scale of the dataset and limitations in computational resources, we restrict our fine-tuning to smaller models with fewer than 1 billion parameters. During training, the input format of decoder is described as: operation: π, premise: x, conclusion: c. During inference, we omit c, allowing the model to generate the conclusion. All models are evaluated on the same testset. Further experimental details are provided in the supplementary material.

To evaluate generation quality, we employ both the BLEU score (Papineni et al. 2002) and accuracy (acc) as performance metrics. For the Math Reasoning task, in addition to the in-distribution test set, we also assess model performance on an out-of-distribution (OOD) test set, where the mathematical expressions are composed using a different set of variables. As shown in Table 1, Qwen2.5 demonstrates consistently strong performance across all tasks. Based on these results, we select Qwen2.5 as the decoder model for subsequent experiments.

VAE model. Next, we evaluate the performance of the proposed VAE setting on downstream reasoning tasks, where the decoder is Qwen2.5-0.5B, the latent dimension is 32 following the same setup as Optimus (Li et al. 2020). In addition, we include results from the base model trained via fine-tuning (denoted as FT) as well as inference using few-shot examples. The input format remains consistent with previous settings, with examples repeated accordingly. For few-shot selection, we employ BM25 to retrieve the most relevant samples.

As illustrated in Table 2, we can observe that injecting the latent space into the query can generally result in better performance compared with base model and other setups. Intuitively, injecting reasoning information into the Query enables the model to more effectively retrieve the stored Value from memory based on the Key **(Finding 1)**. In this setup, models with a trainable prior demonstrate improved performance on mathematical reasoning tasks, but not necessarily on other NLI tasks. This performance gain is attributed to the highly regular and syntactically consistent nature of the target mathematical expressions. In such contexts, retrieving structurally similar examples can effectively enhance model performance by reinforcing pattern recognition.

Furthermore, when the base model is trained using few-shot examples, its performance declines significantly. We observe that the model repeats to generate more examples. To ensure a fair comparison, we did not apply any filtering to remove these redundant outputs **(Finding 2)**.

Additionally, we evaluate the performance of the VAE model trained with and without BM25-based sample selection. In the absence of the relevance function, training samples are randomly drawn from the corpus. In Table 3, integrating BM25-based selection during training yields enhanced performance, indicating the effectiveness of relevance-guided sampling. From a geometric perspective, BM25 retrieves samples that exhibit the highest lexical similarity, effectively selecting the nearest neighbours in the latent space **(Finding 3)**.

Encoding Evaluation

Latent parametric space. As illustrated in Equation 3, by measuring the cosine similarity between gradient vectors associated with different rules, we can quantify the separability between different rule subspaces in the encoder, comparing settings with strong `cls_weight`=1.0 (left) and weak `cls_weight`=0.1 (right) during training. As illustrated in Figure 2, when the classification weight (`cls_weight`) is set to 1.0, most non-diagonal values are close to zero (orthogonality). In contrast, with `cls_weight` set to 0.1, a greater number of non-diagonal elements exhibit higher values (the red colour elements are much more scattered). This observation suggests that explicit supervision facilitates the separation of reasoning rules within the encoder's parameter space **(Finding 4)**.

Latent sentence space. Since rule separation in the parametric space leads to corresponding separation in the feature

Injection	Train Setup	Math Reason		Math Reason (OOD)		Syllogistic Reason		Explanatory Reason
		bleu	acc	bleu	acc	bleu	acc	bleu
Base model	Zero-shot	0.6260	0.3800	**0.1293**	0.0185	0.9014	0.7000	**0.6566**
	Few-shot	0.1596	0.0614	0.1011	0.0057	0.1534	0.0000	0.0765
	Few-shot (FT)	0.1601	0.0557	0.1022	0.0028	0.1712	0.0000	0.0801
kv_add	prior=False	**0.7285**	**0.5285**	0.1055	0.0200	0.4839	0.3400	0.6127
	weight=1.0	0.4986	0.2128	0.0869	0.0157	0.0719	0.0000	0.2049
	weight=0.5	0.4925	0.2814	0.0504	0.0185	0.4870	0.0900	0.2623
kv_mem	prior=False	0.6430	0.4185	0.0985	0.0200	0.4672	0.3300	0.6313
	weight=1.0	0.5255	0.2300	0.0956	0.0185	0.0000	0.0000	0.6022
	weight=0.5	0.6808	0.4857	0.1130	0.0185	0.9452	0.8500	0.5987
query_add	prior=False	0.6501	0.3885	0.1130	**0.0200**	**0.9681**	**0.9200**	**0.6449**
	weight=1.0	0.7262	0.5057	0.1223	**0.0214**	0.4373	0.3300	0.6118
	weight=0.5	**0.8130**	**0.6642**	**0.1441**	0.0185	**0.9461**	**0.8600**	0.6220

Table 2: Quantitative evaluation, where the base model is Qwen2.5-0.5B. Top two values are highlighed in **bold** and **bold**. Prior=False is the setup of VAE without trainable prior. We can observe that query_add leads to the best performance in general.

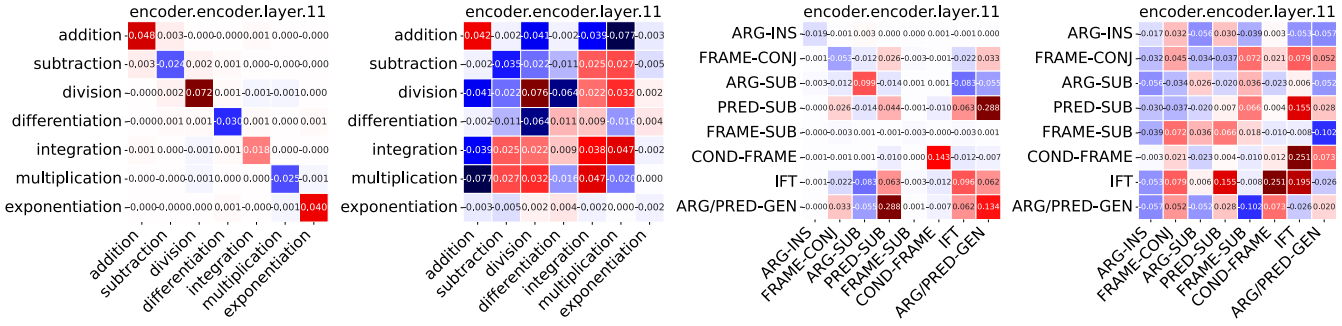

Figure 2: Gradient heatmap for the last posterior encoder layer (query_add setup), where the left two: math derivation, right two: explanatory reasoning. For each task, left is `cls_weight`=1.0, right is `cls_weight`=0.1. We can observe that the non-diagonal values are notably close to 0 when providing higher `cls_weight` (the red colour elements are less scattered), suggesting that incorporating rule information during training enhances the separation of rule subspaces in the encoder's parameter space. We provide the heatmaps of all layers in the supplementary material.

Injection	Prior	BM25	bleu	acc
kv_add	False	False	0.3691	0.0614
kv_mem	False	False	0.5525	0.2371
query_add	False	False	0.5914	0.3021
kv_add	False	True	0.7285	0.5285
kv_mem	False	True	0.6430	0.4185
query_add	False	True	0.6501	0.3885

Table 3: Quantitative evaluation for VAE model with or without BM25 in Math Reasoning task.

space, as shown in Figure 3, the sentence representations tend to form distinct clusters that reflect rule information when the classifier is given a higher `cls_weight`. However, when the `cls_weight` approaches zero, these rule-based clusters disappear. This suggests that the neural network relies more on the memorisation of lexical combination than on rule-based learning **(Finding 5)**.

Case Study for Math Reasoning

Information bottleneck. First, we analyse how varying the number of training samples for each operation affects the reasoning capabilities of the decoder-only LM (Qwen2.5-0.5B). In Figure 4 (left bar plot), it can be observed that when the number of samples per category exceeds 2,000, there is a noticeable decline in accuracy. This suggests that increasing the sample size may introduce greater variability or complexity, potentially disrupting the consistency of each operation. The observation also highlights a limitation of current autoregressive LMs: rather than engaging in rule-based reasoning, they tend to rely on retrieving memorised training instances embedded in their parameters (Zhong et al. 2025). Explicitly injecting latent reasoning representations can help mitigate this issue **(Finding 6)**.

Parametric rule separation. Second, we assess the parametric separation of rule-based information across different model components, specifically the attention (attn) and feed-

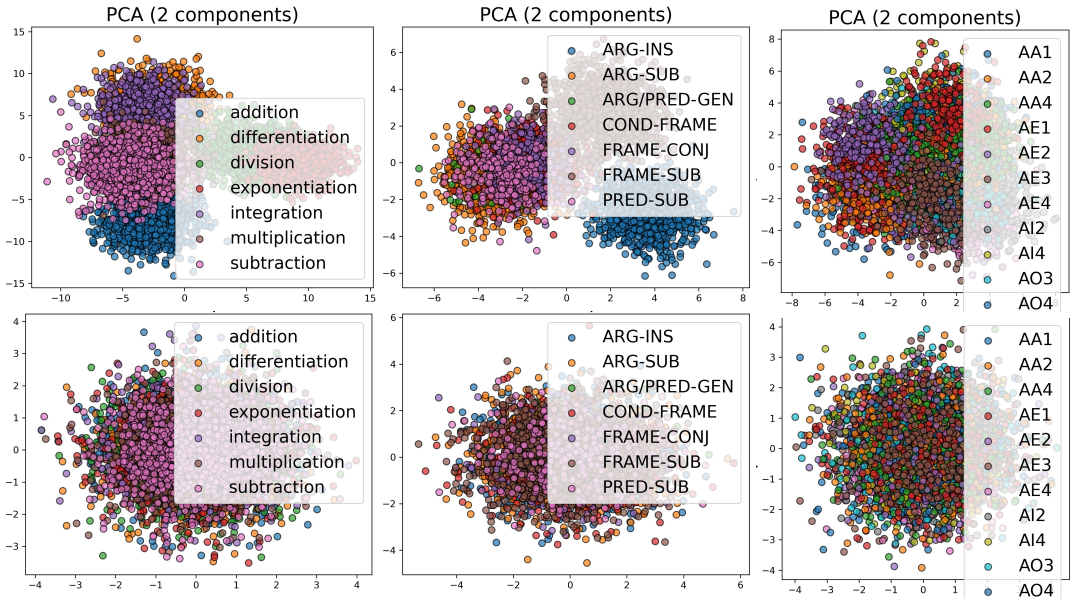

Figure 3: PCA visualisation for `query_add` injection setup, where left three: `cls_weight` is 1.0, right three: `cls_weight` is 0.1. We can observe that the model struggle to learn the rules when the weight is close to zero, indicating the neural network try to deliver reason behaviour via memorisation, rather than rule-based learning. For other injection setups, their visualisations are provided in Figure 11, 12, and 13 in the supplementary material.

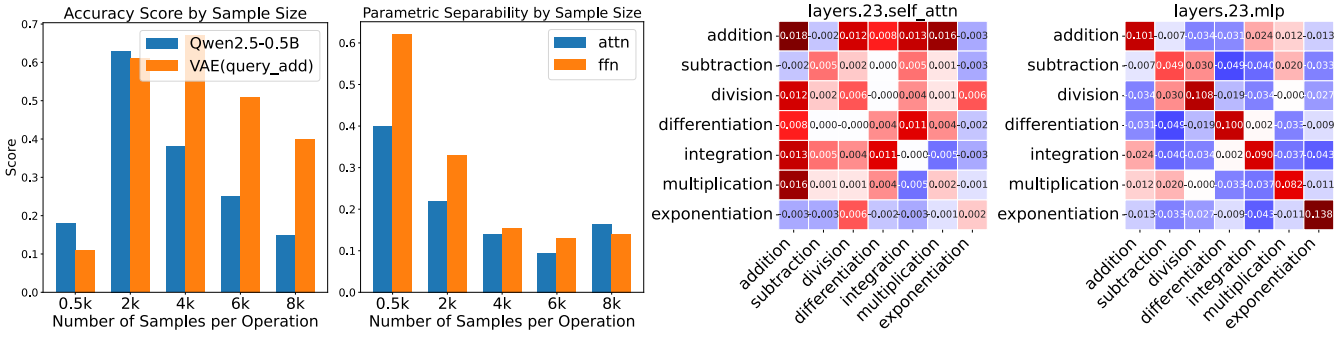

Figure 4: Case study for Math Reasoning task, where left: analysing how varying the number of training samples for each operation affects the reasoning capabilities. Right: comparing the parametric rule separation between attn and ffn at the last layer in Qwen2.5-0.5B, a pretrained checkpoint with a training sample size of 4k.

forward network (FFN) layers, using the same methodology outlined previously. We report the sum of the average diagonal values across all layers. As illustrated in Figure 4 (right bar plot), the FFN layers (right) exhibit a greater tendency to encode rule-based information compared to the attention layers, indicating a more prominent role in capturing structured reasoning patterns **(Finding 7)**.

Conclusion and Future Work

This work serves as a foundational step in exploring rule-based representation learning under the language VAE architecture for NLI tasks. We propose a complete pipeline for learning reasoning rules within Transformer language VAEs. This pipeline encompasses three rule-based reasoning tasks, a supporting theoretical framework, and a practical end-to-end architecture. The experiment illustrates the following findings: **Disentangled reasoning:** Under explicit signal supervision, reasoning rules can be disentangled within the encoder's parametric space, resulting in distinct clustering of rules in the output feature space. **Prior knowledge injection:** injecting reasoning information into the Query enables the model to more effectively retrieve the stored Value from memory based on Key. This approach offers a simple method for integrating prior knowledge into decoder-only language models. In addition, we found that FFN layers are better than attention layers at preserving the separation of reasoning rules in the model's parameters. The future work will focus on the investigation of diffusion-based language models, such as MMaDA (Yang et al. 2025), which can improve flexibility for constraining the decoding process.

Acknowledgements

This work was partially funded by the SNSF project RATIONAL (200021E_229196), the CRUK National Biomarker Centre, and supported by the Manchester Experimental Cancer Medicine Centre and the NIHR Manchester Biomedical Research Centre.

References

Bao, Y.; Zhou, H.; Huang, S.; Li, L.; Mou, L.; Vechtomova, O.; Dai, X.; and Chen, J. 2019a. Generating Sentences from Disentangled Syntactic and Semantic Spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6008–6019.

Bao, Y.; Zhou, H.; Huang, S.; Li, L.; Mou, L.; Vechtomova, O.; Dai, X.-y.; and Chen, J. 2019b. Generating Sentences from Disentangled Syntactic and Semantic Spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6008–6019. Florence, Italy: Association for Computational Linguistics.

Bhattamishra, S.; Patel, A.; Blunsom, P.; and Kanade, V. 2024. Understanding In-Context Learning in Transformers and LLMs by Learning to Learn Discrete Functions. In *The Twelfth International Conference on Learning Representations*.

Bonnet, C.; and Macfarlane, M. V. 2024. Searching Latent Program Spaces. arXiv:2411.08706.

Carvalho, D. S.; Mercatali, G.; Zhang, Y.; and Freitas, A. 2023. Learning Disentangled Representations for Natural Language Definitions. In Vlachos, A.; and Augenstein, I., eds., *Findings of the Association for Computational Linguistics: EACL 2023*, 1371–1384. Dubrovnik, Croatia: Association for Computational Linguistics.

Dalvi, B.; Jansen, P.; Tafjord, O.; Xie, Z.; Smith, H.; Pipatanangkura, L.; and Clark, P. 2021. Explaining answers with entailment trees. *arXiv preprint arXiv:2104.08661*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fang, L.; Zeng, T.; Liu, C.; Bo, L.; Dong, W.; and Chen, C. 2021. Transformer-based Conditional Variational Autoencoder for Controllable Story Generation. arXiv:2101.00828.

Fu, H.; Li, C.; Liu, X.; Gao, J.; Celikyilmaz, A.; and Carin, L. 2019. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 240–250. Minneapolis, Minnesota: Association for Computational Linguistics.

Gu, Y.; Feng, X.; Ma, S.; Zhang, L.; Gong, H.; and Qin, B. 2022. A Distributional Lens for Multi-Aspect Controllable Text Generation. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 1023–1043. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Gu, Y.; Feng, X.; Ma, S.; Zhang, L.; Gong, H.; Zhong, W.; and Qin, B. 2023. Controllable Text Generation via Probability Density Estimation in the Latent Space. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12590–12616. Toronto, Canada: Association for Computational Linguistics.

Hu, Z.; and Li, L. E. 2021. A causal lens for controllable text generation. *Advances in Neural Information Processing Systems*, 34: 24941–24955.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31.

Jansen, P. A.; Wainwright, E.; Marmorstein, S.; and Morrison, C. T. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *arXiv preprint arXiv:1802.03052*.

John, V.; Mou, L.; Bahuleyan, H.; and Vechtomova, O. 2019. Disentangled Representation Learning for Non-Parallel Text Style Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 424–434.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Li, B.; He, J.; Neubig, G.; Berg-Kirkpatrick, T.; and Yang, Y. 2019. A Surprisingly Effective Fix for Deep Latent Variable Modeling of Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3603–3614. Hong Kong, China: Association for Computational Linguistics.

Li, C.; Gao, X.; Li, Y.; Peng, B.; Li, X.; Zhang, Y.; and Gao, J. 2020. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4678–4699.

Liu, G.; Feng, Z.; Gao, Y.; Yang, Z.; Liang, X.; Bao, J.; He, X.; Cui, S.; Li, Z.; and Hu, Z. 2023. Composable Text Controls in Latent Space with ODEs. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 16543–16570. Singapore: Association for Computational Linguistics.

Liu, S.; Xing, L.; and Zou, J. 2023. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.

Meadows, J.; Valentino, M.; Teney, D.; and Freitas, A. 2023. A Symbolic Framework for Systematic Evaluation of Mathematical Reasoning with Transformers. *arXiv preprint arXiv:2305.12563*.

Meadows, J.; Valentino, M.; Teney, D.; and Freitas, A. 2024. A Symbolic Framework for Evaluating Mathematical Reasoning and Generalisation with Transformers. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 1505–1523. Mexico City, Mexico: Association for Computational Linguistics.

Mercatali, G.; and Freitas, A. 2021. Disentangling Generative Factors in Natural Language with Discrete Variational Autoencoders. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Findings of the Association for Computational Linguistics: EMNLP 2021*, 3547–3556. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Morris, J. X.; Sitawarin, C.; Guo, C.; Kokhlikyan, N.; Suh, G. E.; Rush, A. M.; Chaudhuri, K.; and Mahloujifar, S. 2025. How much do language models memorize? arXiv:2505.24832.

Papamarkou, T.; Skoularidou, M.; Palla, K.; Aitchison, L.; Arbel, J.; Dunson, D.; Filippone, M.; Fortuin, V.; Hennig, P.; Hernández-Lobato, J. M.; et al. 2024. Position: Bayesian deep learning is needed in the age of large-scale AI. *arXiv preprint arXiv:2402.00809*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Qwen; :; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Tang, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Shen, T.; Mueller, J.; Barzilay, R.; and Jaakkola, T. 2020. Educating text autoencoders: Latent representation guidance via denoising. In *International Conference on Machine Learning*, 8719–8729. PMLR.

Sun, S.-H.; Noh, H.; Somasundaram, S.; and Lim, J. 2018. Neural Program Synthesis from Diverse Demonstration Videos. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4790–4799. PMLR.

Valentino, M.; Kim, G.; Dalal, D.; Zhao, Z.; and Freitas, A. 2025. Mitigating Content Effects on Reasoning in Language Models through Fine-Grained Activation Steering. arXiv:2505.12189.

Valentino, M.; Meadows, J.; Zhang, L.; and Freitas, A. 2023. Multi-Operational Mathematical Derivations in Latent Space. arXiv:2311.01230.

Valentino, M.; Meadows, J.; Zhang, L.; and Freitas, A. 2024. Multi-Operational Mathematical Derivations in Latent Space. arXiv:2311.01230.

Valentino, M.; Thayaparan, M.; Ferreira, D.; and Freitas, A. 2022. Hybrid autoregressive inference for scalable multihop explanation regeneration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 11403–11411.

van Krieken, E.; Minervini, P.; Ponti, E.; and Vergari, A. 2025. Neurosymbolic Diffusion Models. arXiv:2505.13138.

Vasilakes, J.; Zerva, C.; Miwa, M.; and Ananiadou, S. 2022. Learning Disentangled Representations of Negation and Uncertainty. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8380–8397. Dublin, Ireland: Association for Computational Linguistics.

Yan, Y.; Lu, Y.; Xu, R.; and Lan, Z. 2025. Do PhD-level LLMs Truly Grasp Elementary Addition? Probing Rule Learning vs. Memorization in Large Language Models. arXiv:2504.05262.

Yang, L.; Tian, Y.; Li, B.; Zhang, X.; Shen, K.; Tong, Y.; and Wang, M. 2025. MMaDA: Multimodal Large Diffusion Language Models. arXiv:2505.15809.

Yu, K. L.; Chatzi, E.; and Kissas, G. 2025. Grammar-based Ordinary Differential Equation Discovery. arXiv:2504.02630.

Zhang, Y.; Carvalho, D.; and Freitas, A. 2024. Learning Disentangled Semantic Spaces of Explanations via Invertible Neural Networks. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2113–2134. Bangkok, Thailand: Association for Computational Linguistics.

Zhang, Y.; Carvalho, D.; and Freitas, A. 2025. Quasi-symbolic Semantic Geometry over Transformer-based Variational AutoEncoder. In *Proceedings of the 29th Conference on Computational Natural Language Learning*. Association for Computational Linguistics.

Zhang, Y.; Carvalho, D. S.; Pratt-Hartmann, I.; and Freitas, A. 2024a. Towards Controllable Natural Language Inference through Lexical Inference Types. arXiv:2308.03581.

Zhang, Y.; Valentino, M.; Carvalho, D.; Pratt-Hartmann, I.; and Freitas, A. 2024b. Graph-Induced Syntactic-Semantic Spaces in Transformer-Based Variational AutoEncoders. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Findings of the Association for Computational Linguistics: NAACL 2024*, 474–489. Mexico City, Mexico: Association for Computational Linguistics.

Zhao, T.; Zhao, R.; and Eskenazi, M. 2017. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 654–664. Vancouver, Canada: Association for Computational Linguistics.

Zhong, S.; Xu, M.; Ao, T.; and Shi, G. 2025. Understanding Transformer from the Perspective of Associative Memory. *arXiv preprint arXiv:2505.19488*.