INTERACTION: A Generative XAI Framework for Natural Language Inference Explanations

Jialin Yu*, Alexandra I. Cristea†, Anoushka Harit‡ Zhongtian Sun§
Olanrewaju Tahir Aduragba¶, Lei Shi‖ and Noura Al Moubayed**
Department of Computer Science, Durham University
Durham, UK
Email: {*jialin.yu, †alexandra.i.cristea, ‡anoushka.harit, § zhongtian.sun}@durham.ac.uk
{¶olanrewaju.m.aduragba, ‖lei.shi, **noura.al-moubayed}@durham.ac.uk

***Abstract*—*XAI with natural language processing* aims to produce human-readable explanations as evidence for AI decision-making, which addresses explainability and transparency. However, from an HCI perspective, the current approaches only focus on delivering a single explanation, which fails to account for the *diversity of human thoughts and experiences* in language. This paper thus addresses this gap, by proposing a generative XAI framework, *INTERACTION* (explaIn aNd predicT thEn queRy with contextuAl CondiTional varIational autO-eNcoder). Our novel framework presents explanation in two steps: (*step one*) Explanation and Label Prediction; and (*step two*) Diverse Evidence Generation. We conduct intensive experiments with the Transformer architecture on a benchmark dataset, e-SNLI [1]. Our method achieves competitive or better performance against state-of-the-art baseline models on explanation generation (up to 4.7% gain in BLEU) and prediction (up to 4.4% gain in accuracy) in *step one*; it can also generate multiple diverse explanations in *step two*.**

***Index Terms*—generative model, neural network, deep learning, natural language processing, XAI**

I. Introduction

Traditionally, natural language processing (NLP) applications are built based on techniques that are inherently more explainable. Examples of such techniques are often referred to as *'white box'* techniques, including rule-based heuristic systems, decision trees, hidden Markov models, etc. In recent years, due to the advancement of deep learning, a *'black box'* technique, deep neural network, has become the dominant approach [2]. With the advancement of deep neural networks, their ubiquitousness comes at the expense of less interpretability. Hence, concerns have been raised on whether deep neural networks can make reasonable judgements [3], [4], which further triggers an interest in explainable artificial intelligence (XAI) research [5].

With XAI techniques in NLP applications, researchers first focused on *feature*-based [6], [7], *model*-based [8], and *example*-based [9] explanation techniques. However, even for experts working as data scientists in the industry, interpreting results from these models was found to be hard and bias-prone [10]. To reduce human interpretation bias, directly generating natural language explanations seemed a better medium for presentation. Rather than based on carefully designed additional tools, *XAI with natural language* produced human-readable explanations as evidence for AI decision-making [11].

The current state-of-the-art approaches, such as those in [12], [13], are limited by presenting a single explanation only. However, from an HCI research perspective, it is hard to account for the diversity of human thoughts and experience [14]. Indeed, natural language allows expressing the same semantic content in various 'correct' (i.e., semantically similar) forms, subject to cognitive biases, social expectations, and sociocultural backgrounds [15].

This paper addresses this gap by, proposing a generative XAI framework, which presents explanations in two steps: (step one) Explanation and Label Prediction, and (step two) Diverse Evidence Generation. In step one, we offer the most probable explanation and label prediction, similar to other prior work in literature [12], [13]. In our original step two, we adopt deep generative models, to generate multiple diverse explanations via posterior analysis in the latent space. We evaluate our method specifically on an natural language inference (NLI) task [16], which determines whether a 'hypothesis' is true (entailment), false (contradiction), or undetermined (neutral), given a 'premise'. To perform this, an appropriate dataset is needed. Current NLI datasets, however, contain annotation artefacts, which allow the models to make predictions based on spurious correlations [17]. To address annotation artefacts in data, Camburu *et al.* [1] suggest that spurious correlations are much harder to be captured with natural language explanations and propose a large-scale benchmark dataset (e-SNLI), which contains NLI data points and their associated explanations. In this paper, we present our studies thus on this dataset, with the Transformer architecture, as further explained in Section IV and Section V.

Our main contributions include: *(i)* a novel two-step generative XAI framework, **INTERACTION**, which presents explanations in two steps: (step one) Explanation and Label Prediction; and (step two) Diverse Evidence Generation; *(ii)* the first study on spurious correlation on the e-SNLI dataset with Transformer architecture; *(iii)* demonstrating the benefits of our framework, **INTERACTION**, against state-of-the-art baseline models with empirical experiments; and *(iv)* a solid deep generative model baseline for future research in the XAI field.

II. Related Work

A. Explainable Artificial Intelligence for Natural Language Processing

General XAI approaches can be categorised in two main ways: [18], [19]: 1) Local vs Global, and 2) Self-Explaining vs Post-Hoc. Our work contributes to explainable artificial intelligence (XAI) from two perspectives: *Local* and *Self-Explaining*, as we provide explanations based on fine-granularity individual input, and our explanations are directly interpretable.

In terms of explanation techniques and their applications to NLP there are, in general, five different types [2]: 1) feature importance, 2) surrogate model, 3) example-driven, 4) provenance-based, and 5) declarative induction. The first three are more widely adopted and have already been described briefly in section I. The provenance-based technique refers to visualising some or all of the prediction process, such as in [20], [21]. Our work uses the *declarative induction technique*, which tackles the challenging task of providing human-readable representations as part of the results, such as in [1], [22]. Our work further extends [1] with a *probabilistic treatment*.

B. Supervised Deep Generative Models for Natural Language Processing

Our work is associated with deep generative models, which is based on Neural Variational Inference (NVI) [23]–[25]. NVI is also known as amortised variational inference in the literature and can be considered as an extension of the mean-field variational inference [26], [27]. The NVI technique uses data-driven neural networks instead of more restrictive statistical inference techniques. NVI allows us to infer unobservable latent random variables that generate the observed data and are thus very efficient for data with hidden structures, such as natural language.

NVI has been successfully applied in various NLP applications including topic modelling [28], [29], machine translation [30], [31], text classification [28], conversation generation [32], [33], and story generation [34]. This paper explores the *potential for XAI with natural language inference explanation generation* with a novel deep generative framework. A very recent published paper [35] adopts a similar approach as in this paper, however, the research gap for multiple explanations generation is not explored or discussed. This paper is thus, to the best of our knowledge, the *first work to address the concern on the diversity of human languages* in XAI within the natural language inference task.

III. Technical Background

This section provides a brief overview of the Conditional Variational Autoencoder (CVAE), the Transformer architecture, and a description of the data.

A. Conditional Variational Autoencoder

CVAE [36], [37] is an extended version of the deep generative latent variable model (LVM) based on the variational autoencoder (VAE) model [23], [25]. Both models allow learning rich, nonlinear representations for high-dimensional inputs. When compared with VAE (performing inferences for the latent representation $\boldsymbol{z}$, based on the input $\boldsymbol{x}$, only), CVAE performs inference for the latent representation $\boldsymbol{z}$, based on **both** the input $\boldsymbol{x}$ and the output $\boldsymbol{y}$, together. CVAE can be considered as a neural network framework based on supervised Neural Variational Inference.

CVAE generally includes two components: an encoder and a decoder. We consider the joint probability distribution and its factorisation, in the form of $\boldsymbol{p_\theta(y,z|x) = p_\theta(y|z,x)p_\theta(z|x)}$ as in [28], [31]–[34]. The encoder $\boldsymbol{p_\theta(z|x)}$ takes the observed input $\boldsymbol{x}$ and produces a corresponding latent vector $\boldsymbol{z}$ as the output with parameter θ. The decoder $\boldsymbol{p_\theta(y|z,x)}$ takes the observed input $\boldsymbol{x}$ and its corresponding latent vector sample $\boldsymbol{z}$ as the total input and produces an output $\boldsymbol{y}$ with the parameter θ. The latent variable $\boldsymbol{z}$ in the joint probability $\boldsymbol{p_\theta(y,z|x)}$ can be marginalised out by taking samples from $\boldsymbol{p(z)}$.

For CVAE, we optimise the following evidence lower bound (ELBO) for the log-likelihood during training:

$$\log p_\theta(y|x) \geq \mathcal{L}(ELBO) = E_{q_\phi(z)}[\log p_\theta(y|z,x)] - D_{KL}[q_\phi(z|x,y)||p_\theta(z|x)] \tag{1}$$

The first term of ELBO is the reconstruction loss and is measured via cross-entropy matching between predicted versus real targets $\boldsymbol{y}$. The second term is the Kullback–Leibler (KL) divergence between two distributions $\boldsymbol{p_\theta(z|x)}$ and $\boldsymbol{q_\phi(z|x,y)}$. As the true posterior distribution $\boldsymbol{p_\theta(z|x)}$ is intractable to compute, a variational family distribution $\boldsymbol{q_\phi(z|x,y)}$ is introduced as its approximation. We consider that both $\boldsymbol{p_\theta(z|x)}$ and $\boldsymbol{q_\phi(z|x,y)}$ are in the form of isotropic Gaussian distributions, as $\boldsymbol{\mathcal{N}(\mu_\theta(x), \text{diag}(\sigma^2_\theta(x)))}$ and $\boldsymbol{\mathcal{N}(\mu_\phi(x,y), \text{diag}(\sigma^2_\phi(x,y)))}$. Our work takes a similar assumption, but the key difference lies in the design of our novel model architectures (section V), together with using the Transformer model [38] as a building block.

B. Transformer Architecture

The Transformer architecture, proposed in [38], is the first neural network architecture entirely built upon the self-attention mechanism. It has been used as the main building block for most of the current state-of-the-art models in NLP, such as BERT [39], GPT3 [40], and BART [41]. The Transformer architecture can be divided into three main components: an embedding part, an encoder, and a decoder.

The embedding part takes the input $x \in R^{s_1 \times 1}$ in the form of a sequence with length s_1 and uses an input embedding to create $E(x) \in R^{s_1 \times E}$, where E is the embedded dimension size. Due to the permutation-invariant self-attention mechanism, [38] further introduces positional encoding, to encode sequential order information, as $P(x) \in R^{s_1 \times E}$. The sum of positional encoding and input embedding is used as the final embedding of the input x. In [38], sine and cosine functions of different frequencies are adopted as positional encoding methods. Further work on large-scale transformers [39]–[41]

use a *learnt positional embedding*, which is what we utilise in this paper. For the encoder and the decoder, we use precisely the same Transformer architecture as in the original paper [38]. In our experiments, if an encoder and a decoder are used simultaneously, they each have a separate embedding part.

C. Data Description

Our training data is in the form of N data quadruplets $\{x_n^{(p)}, x_n^{(h)}, y_n^{(l)}, y_n^{(e)}\}_{n=1}^{N}$, with each quadruplet consisting of the *premise* (denoted by $x_n^{(p)}$), the *hypothesis* (denoted by $x_n^{(h)}$) their *associated label* (denoted by $y_n^{(l)}$), and *explanation* (denoted by $y_n^{(e)}$). For the n^{th} quadruplet, $x_n^{(p)} = \{w_1^{(p)}, ..., w_{L_p}^{(p)}\}$, $x_n^{(h)} = \{w_1^{(h)}, ..., w_{L_h}^{(h)}\}$, $y_n^{(l)} = \{w^{(l)}\}$, and $y_n^{(e)} = \{w_1^{(e)}, ..., w_{L_e}^{(e)}\}$ denote the set of L_p words from the premise sentence, L_h words from the hypothesis sentence, a single word $w^{(l)}$ from the label, and L_e words from the explanation sentence, respectively.

Our validation and testing data are similar to data quadruplets as the training data; however, we have three ($y_n^{(e_1)}$, $y_n^{(e_2)}$ and $y_n^{(e_3)}$) instead of one explanation $y_n^{(e)}$, all created by human experts. During training, we update model parameters based on one explanation $y_n^{(e)}$ for n^{th} data entry; and during validation and testing, we perform model selection and inference based on the mean average loss of the three explanations ($y_n^{(e_1)}$, $y_n^{(e_2)}$ and $y_n^{(e_3)}$). In the following, we omit the data quadruplet index n and use bold characters to represent vector form representations, as $\boldsymbol{x^{(p)}}$, $\boldsymbol{x^{(h)}}$, $\boldsymbol{y^{(l)}}$, and $\boldsymbol{y^{(e)}}$.

IV. Preliminary Experiments

We present two preliminary experiments in this section. We use the architecture setting similar to the *base* version of the Transformer model [38], which is a 6-layer model with 512 hidden units and 8 heads for each encoder-decoder network. Based on an inspection of token length statistics (Appendix A), we set the maximum length of 25 for positional encoding. We adopt the pre-processing technique as in [1]. See Appendix C for a detailed description of all model complexity in this paper.

We generally follow the vocabulary processing steps as in [1] (see detailed pre-processing description in Appendix A). We report our quantitative assessment results based on 3 random seeds (1000, 2000, and 3000), and report the average performance with its standard deviation in parenthesis. Regarding quantitative assessment, we use automatic evaluation metrics (Perplexity and BLEU [42]) over the entire test data points. Regarding qualitative assessment (Correct@100, as in Table II and Table III), we report results based on the seed 1000. We adopt the criterion as in [1] and evaluate the Correct@100 score based on the first 100 test examples only[1]. For evaluation, the lower the perplexity, the higher the BLEU score and the higher the Correct@100 score, the better the model performs.

[1]The score is related to the correctness for generated explanation based on the annotations, details described in Appendix B.

We use the maximum a posteriori (MAP) estimate decoding for the conditional generation. MAP decoding, whilst not always the optimal choice, has a reasonably good performance and is widely adopted and cheap to compute [43]. For the network optimisation, we use Adam [44] as our optimiser with default hyperparameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$). We conduct all the experiments with a batch size of 16 and a learning rate of $1e-5$ for a total of 10 epochs on a machine with Ubuntu 20.04 operating system and a GTX 2080Ti GPU.

A. Architecture Selection and Spurious Correlation

In the first experiment, we answer two questions: **Q(i)** *What is a good Transformer model architecture choice for the e-SNLI text classification task?* **Q(ii)** *How easily can a Transformer model pick up the spurious correlation, when only a hypothesis sentence is observed?*

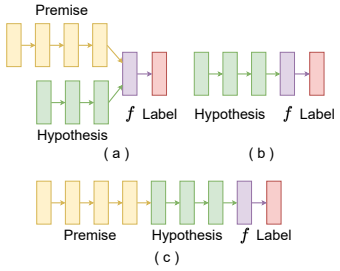

Fig. 1. Graphical overview of architectures used in section IV-A. (a) Separate Transformer Encoder; (b) Premise Agnostic Encoder; and (c) Mixture Transformer Encoder.

TABLE I
Architecture Selection and Spurious Correlation Experiments.

Model	Accuracy (%)
Separate Transformer Encoder	73.97 (0.34)
Mixture Transformer Encoder	78.98 (1.44)
Premise Agnostic Encoder	65.43 (0.72)

To answer **Q(i)**, we experiment on two candidate model architectures: *(1) Separate Transformer Encoder*: an architecture with two separate encoders, one each for the premise and hypothesis sentences, respectively (Fig. 1a). *(2) Mixture Transformer Encoder*: an architecture with a mixture encoder for both premise and hypothesis sentence together (Fig 1c). We choose these two candidates for the following reasons: the first candidate architecture is widely adopted in early NLI literature [45]–[47], where f refers to algorithmic operations (identity, subtraction, multiplication) as in [48]. The latter candidate architecture is adopted by the BERT model [39], where f refers to an affine transformation operation and has achieved state-of-the-art performance for NLI tasks. To answer **Q(ii)**, we perform the premise-agnostic prediction experiment on the *Premise Agnostic Encoder* model (Fig 1b), where f refers to an affine transformation operation.

For the above two experiments, results are presented in Table I. For the *Separate Transformer Encoder*, we use the encoder outputs at two separate '$<$ bos $>$' positions for

algorithmic operations (identity, subtraction, and multiplication). For *Mixture Transformer Encoder* and *Premise Agnostic Encoder*, we use the output at the first '$< bos >$' position. We apply an affine transformation operation for predicting the label. The results suggest the *Mixture Transformer Encoder* outperforms the *Separate Transformer Encoder*, in a statistically significant way ($p < .05$; Wilcoxon test). The *Premise Agnostic Encoder* achieves 82.84% (based on $65.43/78.98$) of the *Mixture Transformer Encoder* performance, suggesting that Transformer models tend to capture spurious correlations very easily for the NLI label prediction task.

B. Premise-Agnostic and Full Generation

In the second experiment, we address two further questions: **Q(iii)** *Is providing explanations as output reducing the impact of spurious correlation in a Transformer model, compared to predicting the label only?* **Q(iv)** *How much better are explanations based on premise and hypothesis together, instead of hypothesis-only?*

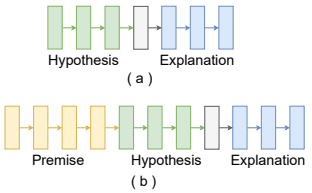

Fig. 2. Graphical overview of architectures used in section IV-B. (a) Agnostic Generation; (b) Full Generation.

TABLE II
PREMISE AGNOSTIC GENERATION EXPERIMENTS.

Model	Perplexity	BLEU	Correct@100
Agnostic Generation	7.66 (0.03)	25.74 (0.8)	42.87
Full Generation	5.53 (0.05)	33.14 (0.5)	57.45

To answer **Q(iii)**, we follow and extend the *'PremiseAgnostic'* experiment [1]. We use the model architecture shown in Fig. 2a, and we are interested in evaluating how well the model can generate an explanation from the premise-agnostic scenario (only premise observed). To answer **Q(iv)**, we implement the seq2seq framework [49] with the Transformer architecture. We compare the agnostic generation scenario with the full generation scenario (both premise and hypothesis observed, as shown in Fig. 2b).

Our results, presented in Table II, suggest that the agnostic generation significantly reduces ($p < .05$; Wilcoxon test) the ability to generate correct explanations, with only 72.19% (based on $5.53/7.66$) for perplexity, 77.67% (based on $25.74/33.14$) for the BLEU score, and 74.62% (based on $42.87/57.45$) for the Correct@100 score (compared to 82.84% in section IV-A).

V. PROPOSED DEEP GENERATIVE XAI FRAMEWORK

In this section, we explain in detail our novel framework, **INTERACTION** - (explaIn aNd predicT thEn queRy with contextuAl CondiTional varIational autO-eNcoder). Our framework presents explanation in two steps: (step one) *Explanation and Label Prediction*; and (step two) *Diverse Evidence Generation*. We present a workflow diagram for our framework in Fig. 3, which consists of four components as follows.

A. Neural Encoder

Given a pair of premise $\boldsymbol{x^{(p)}}$ and hypothesis $\boldsymbol{x^{(h)}}$, with their associated explanation $\boldsymbol{y^{(e)}}$, the encoder network outputs two sequences of representations:

$$\begin{aligned} \boldsymbol{x_h} &= \boldsymbol{Encoder}([\boldsymbol{x^{(p)}}; \boldsymbol{x^{(h)}}]) \\ \boldsymbol{y_h} &= \boldsymbol{Encoder}([\boldsymbol{y^{(e)}}]) \end{aligned} \tag{2}$$

Here $\boldsymbol{Encoder}$ refers to the *Transformer Mixture Encoder*, which is selected based on experiments in section IV-A. $\boldsymbol{x_h}$ is the contextual representations for the premise $\boldsymbol{x^{(p)}}$ and hypothesis $\boldsymbol{x^{(h)}}$ pair. $\boldsymbol{y_h}$ is the contextual representation for explanation $\boldsymbol{y^{(e)}}$. We share the same encoder network parameters for producing $\boldsymbol{x_h}$ and $\boldsymbol{y_h}$. $\boldsymbol{x_h}$ has the same sequence length as the sum of premise and hypothesis length. $\boldsymbol{y_h}$ has the same sequence length as the explanation length. $[\boldsymbol{a}; \boldsymbol{b}]$ refers to the concatenation operation of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

B. Neural Inferer

The neural inferer can be divided into two separate components: the prior and the posterior networks. As determined by the ELBO equation 1, the parameters of the prior are computed by the prior network, which only takes the inputs: $\boldsymbol{x^{(p)}}$ and $\boldsymbol{x^{(h)}}$. The posterior parameters are determined from both inputs and outputs: $\boldsymbol{x^{(p)}}$, $\boldsymbol{x^{(h)}}$ and $\boldsymbol{y^{(e)}}$.

1) Contextual Convolutional Neural Encoder: Before introducing the neural prior and posterior, we first present our novel approach of dealing with various lengths of output from the Transformer encoder. We first adopt the 2d-convolution operations (over the sequence length and hidden dimension) as in [50] and apply it directly to the encoded outputs $\boldsymbol{x_h}$ and $\boldsymbol{y_h}$. For the convolution operations, we use learnable filters with size of 1, 2, and 3 to represent '*unigram*', '*bigram*', and '*trigram*' contextual information from the sequences. Then, we use a max-pooling operation over each filter output, to alleviate various sequence-length issues and concatenate them as one single output vector. Finally, we apply an affine transformation on the output vector and return the original vector dimension, but with a sequence length of 1. We name the whole set of operations here **c**ontextual **co**nvolutional **n**eural en**coder** (denoted in short as $\boldsymbol{Concoder}$).

In contrast, a standard CVAE model uses a fixed position from the sequence instead, to handle various sequence-length issues. We implement a standard CVAE with the $< bos >$ position output as the final output, denoted as *CVAE Generation*. We use this as a comparison with our novel solution ($\boldsymbol{Concoder}$), denoted as *ConCVAE Generation* (with results shown in Table III).

2) Neural Prior: The prior distribution, denoted as:

$$p_\theta(z|x) = \mathcal{N}(z|\mu_\theta(x), \text{diag}(\sigma^2_\theta(x))) \tag{3}$$

$\boldsymbol{p_\theta(z|x)}$ is an isotropic multivariate Gaussian with mean and variance matrices parameterised by neural networks. With variable-length sentence as input, we first use a contextual convolutional neural network, introduced in section V-B1, to retrieve a fixed output $\boldsymbol{x_c}$. Then, we apply two additional affine transformations, $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$, to parameterise the mean and variance matrices for the neural prior. The $\boldsymbol{tanh()}$ function here introduces additional non-linearity and also contributes to numerical stability during parameter optimisation. Thus:

$$\begin{aligned} x_c &= \boldsymbol{Concoder}([x_h]) \\ \mu_\theta &= f_1([x_c]) \\ \log \sigma_\theta &= tanh(f_2([x_c])) \end{aligned} \tag{4}$$

3) Neural Posterior: During training, the latent variable will be sampled from the posterior distribution:

$$q_\phi(z|x, y) = \mathcal{N}(z|\mu_\phi(x, y), \text{diag}(\sigma^2_\phi(x, y))) \tag{5}$$

$\boldsymbol{q_\phi(z|x, y)}$ is also an isotropic multivariate Gaussian with mean and variance matrices parameterised by neural networks. However, the parameters are inferred based on both inputs and outputs. We use the same $\boldsymbol{Concoder}$ network to handle the various lengths of inputs and outputs ($\boldsymbol{x^{(p)}}$, $\boldsymbol{x^{(h)}}$, and $\boldsymbol{y^{(e)}}$). As for the neural prior, we apply two additional affine transformations, $\boldsymbol{f_3}$ and $\boldsymbol{f_4}$, to parameterise the mean and variance matrices. Thus:

$$\begin{aligned} y_c &= \boldsymbol{Concoder}([y_h]) \\ \mu_\phi &= f_3([x_c; y_c]) \\ \log \sigma_\phi &= tanh(f_4([x_c; y_c])) \end{aligned} \tag{6}$$

C. Neural Decoder

The decoder models the probability of the explanation $\boldsymbol{y^{(e)}}$ in an auto-regressive manner, given the predicted label $\boldsymbol{y_p}$, the encoded premise and hypothesis pair $\boldsymbol{x_h}$, and the latent vector $\boldsymbol{z}$. We obtain the explanation sequence via:

$$y^{(e)} = \boldsymbol{Decoder}([z; x_{(h)}]) \tag{7}$$

Here, $\boldsymbol{Decoder}$ refers to the Transformer decoder. Given an explanation with a total sequence length of T, at time step j ($j < T$), it produces the j^{th} word with a softmax selection from the vocabulary based on all the past $j-1$ words.

D. Neural Predictor

In our novel **INTERACTION** framework, the label can be predicted based on one of the three options: *(i)* **M1 Model**: predicted based on the premise and hypothesis only, *(ii)* **M2 Model**: predicted based on the explanation only, and *(iii)* **M3 Model**: predicted based on the premise, hypothesis, and explanation all together. With the Transformer architecture, we first concatenate the vector outputs of the information at each first '$< bos >$' position into a single vector for each model. Then we apply an affine transformation operation $\boldsymbol{f}$ to the concatenated vector. We jointly train the neural predictor together with the generative model *ConCVAE*. We compare the performance of these three models in our experiments (Table III).

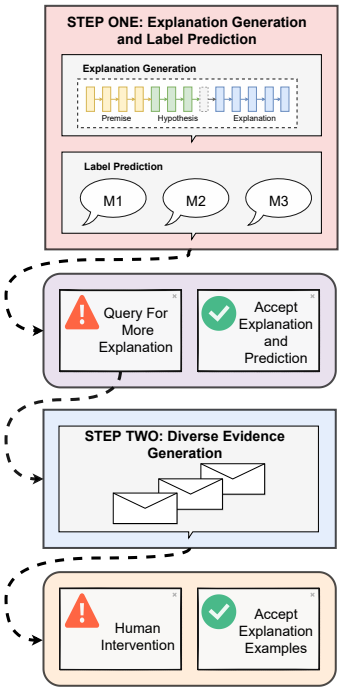

Fig. 3. Graphical overview of our framework, **INTERACTION**, introduced in section V.

VI. Experiments

In this section, to evaluate our proposed framework **INTERACTION**, we conduct experiments comparing with our baseline models.

A. Baseline Models

We define two types of baseline models: *generative model* and *predictive model*. We consider the following works as baseline models:

- seq2seq (*generative model*, our implementation): a sequence-to-sequence learning framework developed by [49]. We implement it with the Transformer architecture and present the results as *Full Generation* in Table III.
- CVAE (*generative model*, our implementation): a strong probabilistic conditional generation framework introduced by [36], [37]. We implement it with the Transformer architecture and present the results as *CVAE Generation* in Table III.
- Transformer (*predictive model*, our implementation): a very strong baseline model for NLI task developed by [38]. We present the results as *Mixture Transformer Encoder* in Table III.

B. Experiment Setup

To evaluate the explanation generative model of our **INTERACTION** framework, we implement our novel *ConCVAE*

TABLE III
XAI WITH NATURAL LANGUAGE PROCESSING RESULTS ('−−' REFERS TO RESULTS NOT APPLICABLE).

Model	Label Accuracy	Perplexity	BLEU	Correct@100
Premise Agnostic Encoder (lower bound)	65.43 (0.72)	−−	−−	−−
Mixture Transformer Encoder (predictive model baseline)	**78.98** (1.44)	−−	−−	−−
Full Generation (generative model baseline, non-probabilistic)	−−	**5.53** (0.05)	**33.14** (0.50)	**57.45**
CVAE Generation (generative model baseline, probabilistic)	−−	7.58 (0.27)	25.70 (1.04)	43.04
ConCVAE Generation (our model, probabilistic)	−−	5.69 (**0.03**)	32.74 (**0.09**)	55.27
INTERACTION M1 (our model)	**83.42 (0.31)**	6.73(0.16)	30.46(0.33)	47.04
INTERACTION M2 (our model)	73.73(1.54)	**5.75 (0.01)**	32.68(0.64)	52.29
INTERACTION M3 (our model)	79.85(0.35)	5.93(0.02)	**32.70 (0.28)**	**58.06**

TABLE IV
SELECTED DIVERSE EVIDENCE GENERATION EXAMPLES.

Test Data Number	29
Premise	a **couple** walk hand in hand down a street .
Hypothesis	the **couple** is **married** .
Explanation	just because the couple is hand in hand does n't mean they are married .
Generated Explanation 1	not all couple walking down street are married .
Generated Explanation 2	not all couple in hand is married .
Generated Explanation 3	not all couples are married .
Test Data Number	50
Premise	a little boy in a gray and white striped sweater and tan pants is **playing** on a piece of **playground equipment** .
Hypothesis	the boy is **sitting on** the school bus on his **way home** .
Explanation	the boy is either playing on a piece of playground equipment or sitting on the school bus on his way home .
Generated Explanation 1	the boy can not be playing on a playground and sitting on his way home at the same time .
Generated Explanation 2	the boy can not be playing on a playground and sitting on his way home simultaneously .
Generated Explanation 3	the boy can not be playing on a playground and sitting on the bus at the same time .
Test Data Number	64
Premise	people jump over **a mountain** crevasse on a rope .
Hypothesis	people are jumping **outside** .
Explanation	the jumping over the mountain crevasse must be outside .
Generated Explanation 1	people jump over a mountain so they must be outside .
Generated Explanation 2	a mountain is outside .
Test Data Number	77
Premise	a man in a black shirt is **looking at** a bike in a workshop .
Hypothesis	a man is deciding which bike **to buy**
Explanation	just because the man is looking at a bike does n't mean he is deciding which bike to buy .
Generated Explanation 1	just because a man is looking at a bike in a workshop does n't mean he is deciding to buy .
Generated Explanation 2	just because a man is looking at a bike in a workshop does n't mean he is deciding what to buy .

model and use the MAP decoding over the latent variable during both training and testing to generate a single explanation. For label prediction task, we implement the **INTERACTION M1**, **M2**, and **M3** models (as in section V-D), and compare their performance with our predictive and generative baseline models. Regarding network architectures, vocabulary, and training, we use the same experimental setting as in section IV.

C. Diverse Evidence Generation via Interpolation

We present a study on the generation of diverse evidence to support explanation, as in *step two* from Figure 3. To generate multiple explanations, we perform posterior analysis over the latent space. We choose to linearly interpolate the isotropic multivariate Gaussians over its 95.44% region (left and right of 2σ from μ). This interpolation produces 5 samples calculated based on the $\mu - 2\sigma$, $\mu - \sigma$, μ, $\mu + \sigma$, and $\mu + 2\sigma$ coordinates. Examples of interpolation results from the *ConCVAE Generation* experiment are presented in Table IV and we only show the examples which are different.

VII. RESULTS AND DISCUSSIONS

A. Explanation Generation Only

The main results are presented in Table III. For the explanation generation evaluation, we first compare a deep generative model (*CVAE Generation*) with a standard neural network model (*Full Generation*). The results suggest that the *Full Generation* model performs better, as the perplexity is reduced by $(7.58 - 5.53 = 2.05)$, the BLEU score increases by $(33.14 - 25.70 = 7.4\%)$, and the Correct@100 score increases $(57.45 - 43.04 = 14.4)$. All the results here are statistically significant ($p < .05$) based on the Wilcoxon signed-rank test. However, deep generative models, such as *CVAE Generation*, allow generating multiple explanations via a posterior analysis over the latent space, as shown in section VI-C. With our novel contextual deep generative model *ConCVAE*, we achieve competitive performance with the *Full Generation* model, evidenced in both quantitative (perplexity, BLEU score) and qualitative (Correct @100) results.

B. Explanation Generation and Label Prediction

We implement three variants of our **INTERACTION** framework (**M1**, **M2** and **M3**) to perform generation and prediction simultaneously. Regarding label prediction, results suggest that generating a valid explanation from the premise and hypothesis sentence-pair allows the encoder to better understand the semantics of the words and hence further enhances the accuracy of prediction. This leads to a boost in prediction performance (83.42% for **M1** and 79.85% for **M3**), compared to the *Mixture Transformer Encoder* (78.98%), with the same number of parameters. However, with **M1**, a significant improvement in classification accuracy results in the worst generation quality (based on Correct@100) among all three models. Additionally, as shown in **M2** model, the label prediction accuracy is the worst when using explanation only. This could potentially be explained since only

52.29% of the explanations are considered as correct (based on Correct@100).

Regarding explanation generation, we observe that the **M3** model achieves competitive results for the quantitative assessment (perplexity and BLEU) as the *Full Generation* model. Additionally, it achieves the best performance in qualitative assessment (Correct@100) amongst all models. The results from Table III suggest that label prediction and explanation generation can complement each other and hence enhance the importance of *XAI with natural languages* in practice. When choosing amongst these three models: for the prediction performance,the **M1** model fits the best; however, for the generation performance, the **M3** model is preferable.

C. Diversity of Explanation

The main contribution in this paper is to build a model (**INTERACTION**) capable of providing multiple explanations, reflecting the diversity in natural languages. The motivation is that a natural language usually works in a way such that humans often provide more than one explanation for their actions, and hence may find systems that reply 'monosyllabically', or too briefly, potentially frustrating, or even non-informative [15], [51]. Still, our approach raises other questions, e.g., do humans have enough time to read multiple explanations? How do they pick the best or most faithful one? In a recent paper [52], the authors propose first to generate multiple paraphrases and then select the most faithful one. In our paper (Fig. 3), we alternatively select the most faithful one based on MAP decoding in *step one* (the maximum likelihood for data), then provide multiple explanations in *step two*. The richness and diversity of the generation of multiple explanations can be observed in Table IV (e.g., for test data number 29 and 64). In practice, the MAP decoding might not offer the best results; however, it is a faithful response from the model, given the context of using a 'data-driven' approach with deep learning.

VIII. Conclusion

Here, we have presented **INTERACTION**, a novel deep generative XAI framework, with explanations in two steps: (1) Explanation and Label Prediction; and (2) Diverse Evidence Generation. INTERACTION is the first study which, to the best of our knowledge, *addresses the concern on the diversity of human languages* in XAI, within the natural language processing task. INTERACTION achieves competitive or better performance against state-of-the-art baseline models on both generation (4.7% improvement in BLEU) and prediction (4.4% improvement in accuracy) tasks. We observe that label prediction and explanation generation can complement each other, which further confirms the benefits of *XAI with natural languages* research in practice.

References

[1] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, "e-snli: Natural language inference with natural language explanations," *arXiv preprint arXiv:1812.01193*, 2018.

[2] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, "A survey of the state of explainable ai for natural language processing," *arXiv preprint arXiv:2010.00711*, 2020.

[3] R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla, and A. Weller, "Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning." International Joint Conferences on Artificial Intelligence, Inc., 2017.

[4] R. Challen, J. Denny, M. Pitt, L. Gompels, T. Edwards, and K. Tsaneva-Atanasova, "Artificial intelligence, bias and clinical safety," *BMJ Quality & Safety*, vol. 28, no. 3, pp. 231–237, 2019.

[5] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[6] N. Voskarides, E. Meij, M. Tsagkias, M. De Rijke, and W. Weerkamp, "Learning to explain entity relationships in knowledge graphs," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 564–574.

[7] F. Godin, K. Demuynck, J. Dambre, W. De Neve, and T. Demeester, "Explaining character-aware neural networks for word-level prediction: Do they discover linguistic rules?" *arXiv preprint arXiv:1808.09551*, 2018.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[9] D. Croce, D. Rossini, and R. Basili, "Auditing deep learning processes through kernel-based explanatory models," 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)*, 2019, pp. 4037–4046.

[10] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan, "Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–14.

[11] S. Wiegreffe and A. Marasovic, "Teach me to explain: A review of datasets for explainable natural language processing," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[12] S. Narang, C. Raffel, K. Lee, A. Roberts, N. Fiedel, and K. Malkan, "Wt5?! training text-to-text models to explain their predictions," *arXiv preprint arXiv:2004.14546*, 2020.

[13] S. Kumar and P. Talukdar, "Nile: Natural language inference with faithful natural language explanations," *arXiv preprint arXiv:2005.12116*, 2020.

[14] L. Aroyo and C. Welty, "Truth is a lie: Crowd truth and the seven myths of human annotation," *AI Magazine*, vol. 36, no. 1, pp. 15–24, 2015.

[15] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial intelligence*, vol. 267, pp. 1–38, 2019.

[16] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *arXiv preprint arXiv:1508.05326*, 2015.

[17] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith, "Annotation artifacts in natural language inference data," *arXiv preprint arXiv:1803.02324*, 2018.

[18] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[19] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[20] M. Zhou, M. Huang, and X. Zhu, "An interpretable reasoning network for multi-relation question answering," *arXiv preprint arXiv:1801.04726*, 2018.

[21] A. Amini, S. Gabriel, P. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi, "Mathqa: Towards interpretable math word problem solving with operation-based formalisms," *arXiv preprint arXiv:1905.13319*, 2019.

[22] N. Pröllochs, S. Feuerriegel, and D. Neumann, "Learning interpretable negation rules via weak supervision at document level: A reinforcement learning approach," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, vol. 1. Association for Computational Linguistics, 2019, pp. 407–413.
[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[24] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *International Conference on Machine Learning*. PMLR, 2014, pp. 1791–1799.
[25] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*. PMLR, 2014, pp. 1278–1286.
[26] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.
[27] C. M. Bishop, "Pattern recognition," *Machine learning*, vol. 128, no. 9, 2006.
[28] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *International conference on machine learning*, 2016, pp. 1727–1736.
[29] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," *arXiv preprint arXiv:1703.01488*, 2017.
[30] J. Su, S. Wu, D. Xiong, Y. Lu, X. Han, and B. Zhang, "Variational recurrent neural machine translation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
[31] A. Pagnoni, K. Liu, and S. Li, "Conditional variational autoencoder for neural machine translation," *arXiv preprint arXiv:1812.04405*, 2018.
[32] T. Zhao, R. Zhao, and M. Eskenazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," *arXiv preprint arXiv:1703.10960*, 2017.
[33] J. Gao, W. Bi, X. Liu, J. Li, G. Zhou, and S. Shi, "A discrete cvae for response generation on short-text conversation," *arXiv preprint arXiv:1911.09845*, 2019.
[34] L. Fang, T. Zeng, C. Liu, L. Bo, W. Dong, and C. Chen, "Transformer-based conditional variational autoencoder for controllable story generation," *arXiv preprint arXiv:2101.00828*, 2021.
[35] Z. Cheng, X. Dai, S. Huang, and J. Chen, "Variational explanation generator: Generating explanation for natural language inference using variational auto-encoder," *International Journal of Computer and Information Engineering*, vol. 15, no. 2, pp. 119–125, 2021.
[36] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, pp. 3483–3491, 2015.
[37] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.
[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[40] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.
[41] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.
[42] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
[43] B. Eikema and W. Aziz, "Is map decoding all you need? the inadequacy of the mode in neural machine translation," *arXiv preprint arXiv:2005.10283*, 2020.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[45] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," *arXiv preprint arXiv:1606.01933*, 2016.
[46] Q. Chen, X. Zhu, Z.-H. Ling, D. Inkpen, and S. Wei, "Neural natural language inference models enhanced with external knowledge," *arXiv preprint arXiv:1711.04289*, 2017.
[47] Y. Gong, H. Luo, and J. Zhang, "Natural language inference over interaction space," *arXiv preprint arXiv:1709.04348*, 2017.
[48] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," *arXiv preprint arXiv:1705.02364*, 2017.
[49] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.
[50] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. [Online]. Available: https://aclanthology.org/D14-1181
[51] J. Zhou and S. Bhat, "Paraphrase generation: A survey of the state of the art," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 5075–5086.
[52] E. Cho, H. Xie, and W. M. Campbell, "Paraphrase generation for semi-supervised learning in nlu," in *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, 2019, pp. 45–54.

APPENDIX A
DATASET STATISTICS

TABLE V
TOKEN LENGTH STATISTICS FOR THE E-SNLI DATASET, ALL NUMBERS ROUND TO INTEGER.

Model	Mean	Median	Standard Deviation	Min	Max
Premise	17	15	7	4	84
Hypothesis	11	10	4	3	64
Explanation	16	15	7	2	189

APPENDIX B
QUALITATIVE EVALUATION

We calculate the qualitative assessment score, Correct@100, as suggested in [1]: we manually grade the correctness of first 100 test examples, each with a score between 0 (incorrect) and 1 (correct) and give partial scores of k/n if only k out of n required arguments were mentioned. The require arguments are publicly available on GitHub[2] and we take the mean average of three annotations as the final score.

APPENDIX C
MODEL COMPLEXITY

We present the model complexity in Table VI, with separate counts for prediction, generation and total network components, the one with the '$--$' mark is denoted as not applicable.

TABLE VI
NUMBER OF PARAMETERS FOR EACH MODEL, WITH SEPARATE COUNTS FOR PREDICTION AND GENERATION COMPONENT.

Model	Prediction	Generation	Total
Separate Transformer Encoder	48.6M	–	48.6M
Mixture Transformer Encoder	24.3M	–	24.3M
Premise Agnostic Encoder	24.3M	–	24.3M
Agnostic Generation	–	63.6M	63.6M
Full Generation	–	63.6M	63.6M
CVAE Generation	–	65.9M	65.9M
ConTrCVAE Generation	–	68.3M	68.3M
INTERACTION M1	24.3M	68.3M	68.3M
INTERACTION M2	24.3M	68.3M	68.3M
INTERACTION M3	24.3M	68.3M	68.3M

[2]https://github.com/OanaMariaCamburu/e-SNLI/tree/master/dataset