World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:2, 2021

Variational Explanation Generator: Generating Explanation for Natural Language Inference using Variational Auto-Encoder

Zhen Cheng, Xinyu Dai, Shujian Huang, Jiajun Chen

Abstract—Recently, explanatory natural language inference has attracted much attention for the interpretability of logic relationship prediction, which is also known as explanation generation for Natural Language Inference (NLI). Existing explanation generators based on discriminative Encoder-Decoder architecture have achieved noticeable results. However, we find that these discriminative generators usually generate explanations with correct evidence but incorrect logic semantic. It is due to that logic information is implicitly encoded in the premise-hypothesis pairs and difficult to model. Actually, logic information identically exists between premise-hypothesis pair and explanation. And it is easy to extract logic information that is explicitly contained in the target explanation. Hence we assume that there exists a latent space of logic information while generating explanations. Specifically, we propose a generative model called Variational Explanation Generator (VariationalEG) with a latent variable to model this space. Training with the guide of explicit logic information in target explanations, latent variable in VariationalEG could capture the implicit logic information in premise-hypothesis pairs effectively. Additionally, to tackle the problem of posterior collapse while training VariaztionalEG, we propose a simple yet effective approach called Logic Supervision on the latent variable to force it to encode logic information. Experiments on explanation generation benchmark—explanation-Stanford Natural Language Inference (e-SNLI) demonstrate that the proposed VariationalEG achieves significant improvement compared to previous studies and yields a state-of-the-art result. Furthermore, we perform the analysis of generated explanations to demonstrate the effect of the latent variable.

Keywords—Natural Language Inference, explanation generation, variational auto-encoder, generative model.

I. Introduction

NATURAL Language Inference (NLI) is a long-standing problem in NLP research, which aims to determine the logic relationship of the given premise-hypothesis pair. As one of the most important natural language understanding task [1], the interpretability of NLI models is important in many applications like the medical and legal scene. Recently, a human-annotated explanation dataset called e-SNLI [2] is proposed for NLI, which yields an extension task of NLI, i.e., explanation generation. Trained on e-SNLI, models could provide logic relationships and explanations of premise-hypothesis pairs simultaneously, which rise the interpretability of final decision in NLI.

According to the order of logic relationship prediction and explanation generation, [2] proposed two paradigms called PredictAndExplain and PredictThenExplain for explanation generation of NLI. In this paper, we follow the latter one for two folds: First, preliminary experiment shows that a simple classifier using human-annotated explanations could obtain an accuracy of 96.83% on the test set of e-SNLI [2], which is much higher than using premise-hypothesis pairs. Second, we can focus more on explanation generation and generate better explanations for interpretability. Like most textual generation tasks, the dominative explanation generation methods [2] are mainly based on discriminative encoder-decoder framework [3]. In this framework, premise and hypothesis are encoded into a fixed-length vector separately and fed into the decoder to generate explanations. Additionally, attention mechanism [4], [5] is used to capture the attentive context of premise-hypothesis pair in each decoding step. However, existing discriminative explanation generators only using premise-hypothesis pairs cannot model this implicitly logical information effectively. As a result, these generators usually generate explanations with correct evidence but incorrect logic semantic. This is because logic information is implicitly contained in the premise-hypothesis pairs and difficult to model. On the other hand, logic information is explicitly expressed in the target explanations and easy to extract. Considering the identical logic information shared between premise-hypothesis pair and explanation, we assume that there exists a latent space of this logic information. To model this latent space, we propose a deep generative model called Variational Explanation Generator (VariationalEG) using Conditional Variational Auto-Encoder (CVAE) [6]. Specifically, a continuous latent variable is introduced to model the identical logic information shared between premise-hypothesis pair and explanation. In addition to use the source sentence pairs, decoder in VariationalEG generates explanations with this logic latent variable. Nevertheless, the proposed VariationalEG is non-trivial to train. Like most existing variational text generation models [7], [8], [9], the proposed VariationalEG encounters posterior collapse issue while optimizing the Evidence Lower-bound (ELBO) [10]. Considering that logic information can be expressed in two forms—logic relationships of label and logic semantic of explanation, we introduce an additional loss called Logic Supervision on the latent variable to predicte logic relationships by latent variable. Experiments on e-SNLI [2] demonstrate that this simple yet effective approach tackles the posterior collapse in explanation generation.

Zhen Cheng, Xinyu Dai*, Shujian Huang, and Jiajun Chen are with the State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing, Jiangsu, China (*corresponding author, e-mail: daixinyu@nju.edu.cn).

Finally, the proposed VariationalEG achieves significant improvement compared to our discriminative base model, i.e., Transformer-based Explanation Generator (TransformerEG). Also, VariationalEG yields a state-of-the-art result compared to previous studies. We analyze the generated explanations of these methods and demonstrate the effect of VariationalEG.

II. Related Work

A. Explanation Generation for Natural Language Inference

As one of the most important natural language understanding tasks [1], Natural Language Inference (NLI) [11] should not be limited to predict logic relationships between premise and hypothesis only. Thanks to the release of a large human-annotated explanation corpus for NLI [2], i.e. e-SNLI, a new domain called explanation generation attracts much attention due to its interpretability in logic relationships prediction. In e-SNLI, models do not only need to predict logic relationships, also generate the explanations. So it suffers "the chicken or the egg" issue. That is to say, whether to generate explanations first or predicate logic relationships first. [2] proposed two paradigms called PredictAndExplain and PredictThenExplain:

- PredictAndExplain: PredictAndExplain is a joint training architecture, in which logic relationships prediction and explanation generation are performed simultaneously. Specifically, logic relationship is inserted as an logic word in the front of explanation. Before generating explanations, PredictAndExplain predicts the logic relationships by generating the corresponding logic word first, i.e., {Entailment,Neutral,Contradiction}.
- PredictThenExplain: PredictThenExplain believes that better explanation could enhance the performance of logic relationship prediction. Specifically, PredictThenExplain trains two sub-models for explanation generation and logic relationship prediction separately. And the logic relationship is predicted using the generated explanation.

Our VariationalEG focuses on generating better explanations which can be considered as the explanation generation part of PredictThenExplain. To compare the performance of logic relationships prediction with other PredictThenExplain models justly, we use the same classifier as [2]. Explanation-based logic relationship prediction could be improved as future work.

B. Variational Auto-Encoder

Variational Auto-Encoder (VAE) [10], [12] has achieved remarkable performance on various text generation tasks such as machine translation [13], dialog generation [14], [15] and other text generation tasks [7], [16]. Instead of being encoded into a fixed point as auto-encoder, source input in VAE is encoded into a distribution. Sampling from this distribution, VAE forms a latent variable and uses it to reconstruct the original input. Based on VAE, [6] proposed an advanced generative model called Conditional VAE (CVAE).

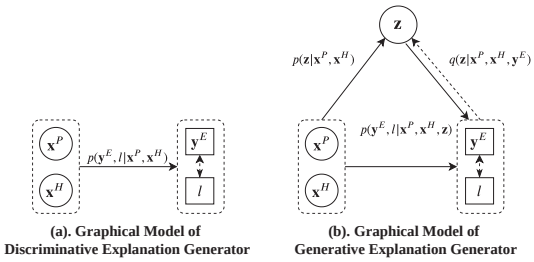

Fig. 1 Graphical models of discriminative explanation generator and generative explanation generator

CVAE could generate different text conditioned on the given context [17], [9]. Obviously, we can employ CVAE to generate explanations conditioned on the premise-hypothesis pairs. However, both VAE and CVAE on text generation are non-trivial to train—KL term in ELBO can easily vanish to 0 and the sampled latent variable becomes non-informative, which is known as *posterior collapse*. To tackle this issue, various approaches [7], [8], [15] have been proposed. Similar to BOW loss [9], Logic Supervision adds an additional loss on latent variable. However, this approach is designed based on the property of explanation generation, i.e., to force latent variable to encode the global indentical logic information between premise-hypothesis pairs and target explanations.

III. Variational Explanation Generator

Notation We denote premise as $\mathbf{x}^P = \langle x_1^P, x_2^P, ..., x_m^P \rangle$ and hypothesis as $\mathbf{x}^H = \langle x_1^H, x_2^H, ..., x_n^H \rangle$, where x_i^P or x_j^H is a token in premise or hypothesis. The corresponding explanation is denoted as $\mathbf{y}^E = \langle y_1^E, y_2^E, ..., y_e^E \rangle$ and the logic relationship is denoted as l. We use $\mathbf{z}$ to present the latent variable.

Fig. 1 shows the graphical models of discriminative explanation generator and generative explanation generator. As shown in Fig. 1 (a), discriminative explanation generation can be formulated as $p(\mathbf{y}^E|\mathbf{x}^P,\mathbf{x}^H)$ and logic relationship is predicted using generated explanation $p(l|\mathbf{y}^E)$. In generative explanation generator (see Fig. 1 (b)), explanation is generated by introducing a latent variable $\mathbf{z}$: $p(\mathbf{y}^E|\mathbf{x}^P,\mathbf{x}^H,\mathbf{z})$. Specifically, latent variable is sampled from a distribution conditioned on premise and hypothesis: $p(\mathbf{z}|\mathbf{x}^P,\mathbf{x}^H)$. The entire generative explanation generation can be written as:

$$\begin{aligned} p(\mathbf{y}^E|\mathbf{x}^P,\mathbf{x}^H) &= \int p(\mathbf{y}^E,\mathbf{z}|\mathbf{x}^P,\mathbf{x}^H)d\mathbf{z}, \\ &= \int p(\mathbf{y}^E|\mathbf{x}^P,\mathbf{x}^H,\mathbf{z})p(\mathbf{z}|\mathbf{x}^P,\mathbf{x}^H)d\mathbf{z}. \end{aligned} \tag{1}$$

According to the evidence lower bound (ELBO) shown in [6], the objective function of generative explanation generator can be derived from (1):

$$\begin{aligned} \log p(\mathbf{y}^E|\mathbf{x}^P,\mathbf{x}^H) &\geq -\mathrm{KL}(q(\mathbf{z}|\mathbf{x}^P,\mathbf{x}^H,\mathbf{y}^E)\|p(\mathbf{z}|\mathbf{x}^P,\mathbf{x}^H)) \\ &\quad + \mathbb{E}_q[\log p(\mathbf{y}^E|\mathbf{x}^P,\mathbf{x}^H,\mathbf{z})] \\ &= \mathcal{L}_{\mathrm{ELBO}}, \end{aligned} \tag{2}$$

which is called as ELBO. By optimizing ELBO, generative explanation generator can be trained efficiently using Stochastic Gradient Variational Bayes (SGVB) estimator [10].

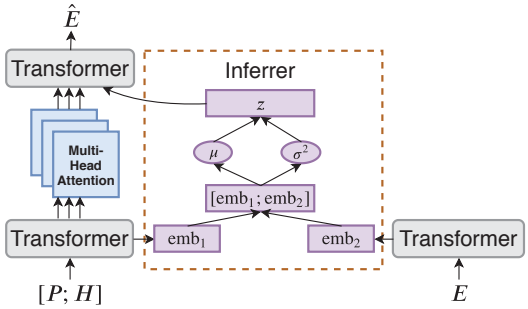

Fig. 2 Overall architecture of the proposed Variational Explanation Generator

The proposed Variational Explanation Generator (VariationalEG) is a generative model. Fig. 2 depicts the overall architecture of VariationalEG, which is mainly composed by three modules: First, a Variational Encoder aims to obtain contextual representations of premise-hypothesis pair and target explanation. Then a Variational Inferrer is responsible for modeling the *prior distribution* $p(\mathbf{z}|\mathbf{x}^P, \mathbf{x}^H)$ and approximating the *posterior distribution* $p(\mathbf{z}|\mathbf{x}^P, \mathbf{x}^H, \mathbf{y}^E)$ in ELBO. Finally, an Attentive Variational Decoder generates explanations by maximizing the probability $p(\mathbf{y}^E|\mathbf{x}^P, \mathbf{x}^H, \mathbf{z})$. In the following of this section, we will introduce these modules in detail.

A. Transformer-Based Variational Encoder

Since the input of explanation generation is a premise-hypothesis pair, previous studies [2] use a siamese RNN-based encoder to obtain the contextual representations of them separately. This structure requires decoder performing twice attentive operations to the source field, which causes attention confliction. That is to say, decoder cannot distinguish to attend premise or hypothesis more at one step.

Inspired by the concatenated input used in BERT [18], we first introduce to use Transformer [19] as explanation generator's encoder. In Transformer-based variational encoder, premise and hypothesis are concatenated as one sequence input with a separation symbol "$\langle$SEP$\rangle$": $\mathbf{x} = [\mathbf{x}^P, \langle\text{SEP}\rangle, \mathbf{x}^H]$. Additionally, we use segment embeddings $\mathbf{x}_{\text{seg}}$ to help model distinguish premise and hypothesis, where premise is denoted as 0 and hypothesis is denoted as 1. The final input of variational encoder from source sentence pair is $\hat{\mathbf{x}} = \mathbf{x} + \mathbf{x}_{\text{seg}}$. Then variational encoder uses Transformer to obtain the contextual representations of the input sentence pairs:

$$\begin{aligned}\mathbf{s}^1 &= \text{TransformerEncoder}([\mathbf{x}^P, \langle\text{SEP}\rangle, \mathbf{x}^H] + \mathbf{x}_{\text{seg}}), \\ &= \langle s_1^1, s_2^1, ..., s_{m+n+1}^1 \rangle.\end{aligned} \tag{3}$$

Moreover, Self-Attention in Transformer models the interaction between premise and hypothesis, which is lacking in the siamese RNN-based encoder.

In addition to encode source premise-hypothesis pair, variational encoder also needs to obtain the contextual representation of target explanation $\mathbf{y}^E$, which will be used in variational inferrer: $\mathbf{s}^2 = \langle s_1^2, s_2^2, ..., s_l^2 \rangle$. All the segment embeddings are set as 0 while encoding the target explanation.

B. Variational Inferrer

Variational inferrer is the core module of VariationalEG, which aims to estimate the premise-hypothesis-related conditional prior distribution $p(\mathbf{z}|\mathbf{x}^P, \mathbf{x}^H)$ and approximate the true explanation-related posterior distribution $p(\mathbf{z}|\mathbf{x}^P, \mathbf{x}^H, \mathbf{y}^E)$. Additionally, variational inferrer samples from conditional prior distribution to obtain latent variable $\mathbf{z}$, which will be used in decoder to generate explanation. To model these two distributions of the latent variable, we introduce two separate networks called *Prior Network* and *Posterior Approximation Network*.

1) Prior Network: Prior Network is in charge of estimating the conditional prior distribution $p(\mathbf{z}|\mathbf{x}^P, \mathbf{x}^H)$ of latent variable $\mathbf{z}$. Specifically, we assume that latent variable $\mathbf{z}$ follows multivariate Gaussian distribution $\mathcal{N}(\mu_1, \sigma_1^2)$ with a diagonal covariance matrix. Based on this assumption, Prior Network estimates mean μ_1 and variance σ_1^2 conditioned on premise-hypothesis pair. First, we use average-pooling of the contextual representations $\mathbf{s}^1$ to obtain the fixed-length vector of premise-hypothesis pair:

$$\mathbf{s}_{\text{avg}}^1 = \frac{1}{m+n+1} \sum_{i=1}^{m+n+1} s_i^1. \tag{4}$$

Then two distinct single layer feed-forward networks (FFN) are used to estimate the mean and variance of prior distribution:

$$\mu_1 = f_{\mu_1}(\mathbf{s}_{\text{avg}}^1), \quad \log \sigma_1^2 = f_{\sigma_1^2}(\mathbf{s}_{\text{avg}}^1), \tag{5}$$

where μ_1 and $\sigma_1^2 \in \mathbb{R}^{d_z}$, and d_z is the dimension of latent variable. Sampling from the premise-hypothesis-related prior distribution, Prior Network provides the latent variable $\mathbf{z}$:

$$\mathbf{z} = \mu_1 + \sigma_1 \odot \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{6}$$

2) Posterior Approximation Network: Since the true posterior distribution $p(\mathbf{z}|\mathbf{x}^P, \mathbf{x}^H, \mathbf{y}^E)$ is intractable to model, we use a Posterior Approximation Network to approximate it depending on target explanation additionally. In this procedure, explicit logic information in explanation is encoded into the posterior distribution of latent variable. By minimizing the KL term in (2), the prior distribution could also encode the explicit logic information. Specifically, Posterior Approximation Network estimates the mean and variance of posterior distribution $\mathcal{N}(\mu_2, \sigma_2^2)$. Similar to the embeddings of premise-hypothesis pair obtained in (4), we first obtain a fixed-length representation of target explanation using average-pooling:

$$\mathbf{s}_{\text{avg}}^2 = \frac{1}{l} \sum_{i=1}^{l} s_i^2. \tag{7}$$

Then mean μ_2 and variance σ_2^2 of posterior distribution are estimated based on premise-hypothesis pair and target explanation:

$$\mu_2 = f_{\mu_2}(\mathbf{s}_{\text{avg}}^1, \mathbf{s}_{\text{avg}}^2), \quad \log \sigma_2^2 = f_{\sigma_2^2}(\mathbf{s}_{\text{avg}}^1, \mathbf{s}_{\text{avg}}^2), \tag{8}$$

where μ_2 and $\sigma_2^2 \in \mathbb{R}^{d_z}$, and $f_{(\cdot)}$ are distinct single layer FFNs.

C. Attentive Variational Decoder

Like most attentive decoders [4], [5] used in text generation, VariationalEG employs Transformer with multi-head attention [19] as decoder to generate explanations. At each step, in addition to depend on the last generated token y^E_{i-1} and the context vector $\mathbf{c}_i$ by attending to premise-hypothesis pair, variational decoder also takes advantage of latent variable $\mathbf{z}$ provided by variational inferrer to generate explanations:

$$\begin{aligned} p(\mathbf{y}^E|\mathbf{x}^P,\mathbf{x}^H,\mathbf{z}) &= \prod_{i=1}^{e} p(y^E_i|y^E_{<i},\mathbf{x}^P,\mathbf{x}^H,\mathbf{z}), \\ &= \prod_{i=1}^{e} g(y^E_{i-1},\mathbf{c}_i,\mathbf{z}). \end{aligned} \tag{9}$$

D. Model Optimizing

The proposed VariationalEG can be trained by optimized the $\mathcal{L}_{\text{ELBO}}$ in (2) using Stochastic Gradient Variational Bayes (SGVB) [10]. However, like most existing VAE-based text generators, VariationalEG faces the challenge to encode meaningful information in the latent variable, which also known as posterior collapse. To tackle this issue, we introduce two approaches to train VariationalEG: KL Annealing and Logic Supervision.

- KL Annealing: KL Annealing is a popular approach proposed in [20] and has been demonstrated its effort in many VAE-based text generation models [9], [16]. The basic idea of KL Annealing is that the weight of KL term in (2) is increased gradually from 0 to 1 during training.
- Logic Supervision: The logic relationship and logic semantic can be treated as two different expressions of logic information. Inspired by this, we propose a simple approach called Logic Supervision on the latent variable. The idea of Logic Supervision is to predict logic relationship using latent variable: $\log p(l|\mathbf{z}) = f(\mathbf{z})$, where f is a multi-layer perceptron.

Finally, the modified objective function with Logic Supervision can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{ELBO}} + \alpha \cdot \mathbb{E}_q \log p(l|\mathbf{z}), \tag{10}$$

where α is the weighting hyper-parameter of Logic Supervision on latent variable.

IV. Experiments

A. Experiments Setup

1) Dataset: We use the benchmark of explanation generation of NLI, i.e., e-SNLI [2], to evaluate the proposed VariationalEG and other explanation generators. e-SNLI contains 570k data points derived from SNLI [20]. Each data point in e-SNLI consists of premise, hypothesis, logic relationship and three explanations with identical logic semantic. We use the standard split of training/development/test datasets with 549,367/9,842/9,824 data points.

TABLE I
Performance of the Proposed VariationalEG and Previous Studies on e-SNLI

Models	Label Accuracy$^{\uparrow}$	Perplexity$^{\downarrow}$	BLEU$^{\uparrow}$	ExplCorrect@100$^{\uparrow}$
e-InferSent†	83.96	10.58	22.40	34.68
Seq2Seq†	81.59	8.95	24.14	49.8
Attention†	81.71	6.1	27.58	64.27
TransformerEG‡	81.83	4.30	31.96	67.6
VariationalEG‡	**85.79**	**4.23**	**32.57**	**75.2**

The higher$^{\uparrow}$ (or the lower $^{\downarrow}$) value means the better performance. † denotes the results are drawn from [2]. ‡ denotes the results are averaged from five experiments with different random seeds.

2) Evaluation Metrics: Following [2], we use four metrics to evaluate the performance of the proposed explanation generator: Label Accuracy, Perplexity, BLEU, and ExplCorrect@100. Perplexity and BLEU [21] are general evaluation metrics commonly used in text generation tasks. Here we take a brief introduction of Label Accuracy and ExplCorrect@100:

- Label Accuracy: In PredictThenExplain, Label Accuracy is obtained using the generated explanations. In PredictAndExplain Label Accuracy is obtained the given premise-hypothesis pairs.
- ExplCorrect@100: Since it is hard to evaluate the correctness of generated explanations only using automatic metrics, we use a human evaluation metric called ExplCorrect@100 to perform on the first 100 data points as supplementary, which focus on the correctness of logic semantic with proper evidence in generations.

B. Experiments Results

Table I shows the overall performance of the proposed VariationalEG and related work on e-SNLI. First, we list the comparable state-of-the-arts models:

- e-InferSent [2]: e-InferSent is an RNN-based PredictAndExplain model, which predicts logic relationships and generates explanations simultaneously. Specifically, logic relationship is inserted as a logic word, i.e., {Entailment,Neutral,Contradiction}, in the front of explanation. Before generating explanations, e-InferSent predicts the logic relationships by generating the corresponding logic word first.
- Seq2Seq [2]: Seq2Seq is an RNN-based PredictThenExplain model, which uses Sequence-to-Sequence [3] model to generate explanations and a BiLSTM with MLP-classifier to predict logic relationships using the generated explanations.
- Attention [2]: Attention adds two separate attention modules in Seq2Seq to performs premise-related attention and hypothesis-related attention at each decoding step. These two attentive contexts are concatenated as input at each step of generating explanations.

Since the proposed VariationalEG is a Transformer-based PredictThenExplain model, we set vanilla Transformer explanation generator (TransformerEG) as baseline.

TABLE II
THE GENERATED EXPLANATIONS OF E-SNLI TEST SET

Premise	A snowboarder on a wide plain of snow.	
Hypothesis	A snow field with a snowboarder on it.	
Models	**Explanation**	**Label**
Groud Truth	wide plain of snow is a rephrasing of snow field.	Entailment
ATTENTION	snow is not the only thing that snow can be plain.	Neutral
TransformerEG	a plain of snow is a snow field.	Entailment
VariationalEG	a plain of snow is a snow field.	Entailment
Premise	A man in a black shirt overlooking bike maintenance.	
Hypothesis	A man watches bike repairs.	
Models	**Explanation**	**Label**
Groud Truth	bike repairs and bike maintenance are nearly synonymous.	Entailment
ATTENTION	a man who is overlooking bike maintenance is not necessarily watching it.	Neutral
TransformerEG	a man overlooking bike maintenance is not necessarily watching bike repairs.	Neutral
VariationalEG	overlooking bike maintenance is the same as watches bike repairs.	Entailment
Premise	A person dressed in a dress with flowers and a stuffed bee attached to it, is pushing a baby stroller down the street.	
Hypothesis	An old lady pushing a stroller down a busy street.	
Models	**Explanation**	**Label**
Groud Truth	a person in a dress of a particular type need neither be old nor female; a street need not be considered busy if only one person is pushing a stroller down it.	Neutral
ATTENTION	the person pushing the baby stroller is not necessarily old.	Neutral
TransformerEG	not all people are old.	Neutral
VariationalEG	not all people are old ladies; not all streets are busy.	Neutral

As shown in Table I, compared to the RNN-based PREDICTTHENEXPLAIN models, TransformerEG achieves substantial improvement on Perplexity, BLEU and ExplCorrect@100. To obtain comparable Label Accuracy results, we reload the classifier used in SEQ2SEQ and ATTENTION to predict logic relationships. Like most RNN-based PREDICTTHENEXPLAIN models, Label Accuracy of TransformerEG still lags behind the PREDICTANDEXPLAIN model, i.e., e-INFERSENT. We find that PREDICTANDEXPLAINs always get better performance on Label Accuracy than PREDICTTHENEXPLAINs, but worse performance on other evaluation metrics. It may be related to the different input—PREDICTTHENEXPLAINs predict logic relationships based on premise-hypothesis pairs and PREDICTANDEXPLAINs predict logic relationships based on the generated explanations, which shows that generated explanations still do not contain enough logic information.

As shown in Table I, the proposed VariationalEG achieves the best performance among the related explanation generators and yields a new state-of-the-art result. Specifically, VariationalEG obtains about 2% improvement on Label Accuracy compared to e-INFERSENT. On Perplexity and BLEU, VariationalEG obtains substantial improvement above TransformerEG. On ExplCorrect@100 of the generated explanation, VariationalEG obtains significant improvement from 64.27 to 75.2. These significant improvements on e-SNLI demonstrate that the proposed VariationalEG does make sense.

C. Experiments Analysis

To verify the motivation of the proposed VariationalEG, we conduct in-depth analysis of the generated explanations, ablation study of Logic Supervision and latent variable.

1) Analysis on Generated Explanations: Tabel II shows three samples from e-SNLI test set and the corresponding generated explanations from the state-of-the-art models.

- The first block in Table II is an entailment sample. Due to the lack of interaction between premise and hypothesis, ATTENTION cannot capture entailment-related parts in premise and hypothesis, i.e., *plain of snow* and *snow field* . As a result, ATTENTION generates neutral explanation with incorrect evidence. Since we concatenate premise and hypothesis as a single input and use the Self-Attention mechanism in Transformer to model the interaction between sentence pairs, both two Transformer-based explanation generators generate correct explanations with entailment-related parts. These generated results show the advantage of Transformer-based explanation generators.
- The second block in Table II is another entailment sample. All three explanation generators capture entailment-related parts in premise-hypothesis pair, i.e., *overlooking bike maintenance* and *watches bike repairs* . However, both ATTENTION and TransformerEG generate explanations with incorrect logic. In contrast, VariationalEG generates a logic-correct explanation due to the effect of the latent variable. Obviously, the proposed VariationalEG can capture better logic information than TransformerEG.
- The third block in Table II is a neutral sample. All three explanation generators generate logic-correct explanations. However, the generated explanations by ATTENTION and TransformerEG only contain part information of complete explanation. These two explanations just point out *old lady* in hypothesis having

TABLE III
ABLATION STUDY OF LOGIC SUPERVISION ON E-SNLI

Models	Label Accuracy↑		Perplexity↓		BLEU↑	
	Dev.	Test.	Dev.	Test.	Dev.	Test.
TransformerEG	81.91	81.83	4.30	4.30	32.84	31.96
TransformerEG (+$\alpha\mathcal{L}_{\text{Logic}}$)	82.61	82.47	4.28	4.29	32.87	32.13
VariationalEG ($\mathcal{L}_{\text{ELBO}}$)	82.83	83.00	4.29	4.30	32.90	32.15
VariationalEG ($\mathcal{L}_{\text{ELBO}} + \alpha\mathcal{L}_{\text{Logic}}$)	85.44	85.79	4.23	4.23	33.26	32.57

TABLE IV
THE GENERATED EXPLANATIONS WITHOUT LOGIC SUPERVISION

Premise	A snowboarder on a wide plain of snow.	
Hypothesis	A snowboarder gliding over a field of snow.	
Models	**Explanation**	**Label**
Groud Truth	just because a snow boarder is on snow does not mean that he is in motion gliding over the snow.	Neutral
VariationalEG ($\mathcal{L}_{\text{ELBO}}$)	gliding over a field of snow is a rephrasing of on a wide plain of snow.	Entailment
VariationalEG ($\mathcal{L}_{\text{ELBO}} + \alpha\mathcal{L}_{\text{Logic}}$)	a snowboarder on a wide plain of snow does not imply gliding over a field of snow.	Neutral

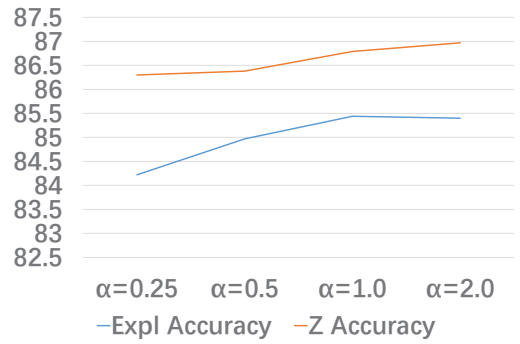

Fig. 3 Effect of weight α on Label Accuracy

no reference in the premise. Since the latent variable could capture global logic information, VariationalEG generates an explanation with complete information. Besides the *old lady* , the explanation points out that the *busy street* in hypothesis is unfounded from the premise. Analyzing on these generated explanations, we have verified the proposed VariationalEG could generate better explanations with correct logic and complete information.

2) Ablation Study of Logic Supervision: Table III shows the ablation study of Logic Supervision on e-SNLI development set and test set. After removing Logic Supervision, Label Accuracy of VariationalEG decreases to 82.83% and 83.00% on development set and test set. Moreover, Perplexity and BLEU of VariationalEG ($\mathcal{L}_{\text{ELBO}}$) degrade to the performance of TransformerEG closely, which is related to the posterior collapse issue. It obviously demonstrates the effect of the proposed Logic Supervision. Additionally, we introduce Logic Supervision on the hidden states of TransformerEG (TransformerEG+$\alpha\mathcal{L}_{\text{Logic}}$) and obtain substantial improvement. But its performance still lags behind VariationalEG, which verifies the effect of the advanced variational generator. To further analyze the effect of Logic Supervision, we show the generated explanations in Tabel IV. Although VariationalEG without Logic Supervision captures the neutral-related parts in the premise-hypothesis pair, i.e., *gliding over the snow* , it generates an explanation with incorrect logic—*a rephrasing of* . Training with Logic Supervision, VariationalEG generates neutral explanation with correct evidence. It demonstrates that Logic Supervision does help the latent variable encode global logic information.

3) Predict Logic Relationship using Latent Variable: Furthermore, we investigate the effect of latent variable $\mathbf{z}$ on Label Accuracy. We denote Label Accuracy using generated explanations as Acc_{expl} and Label Accuracy using latent variable as $\text{Acc}_{\mathbf{z}}$. As shown in Fig. 3, $\text{Acc}_{\mathbf{z}}$ is always higher than Acc_{expl} under different α. Analyzing these 2% relation-explanation inconsistent samples, we find that although latent variable encodes correct logic information, decoder still generates explanations with incorrect logic. It will be a research point of enhancing the influence of the latent variable to decoder in future work.

TABLE V
TIME CONSUMPTION OF TRAINING/INFERENCE ON E-SNLI PER EPOCH

	ATTENTION	TransformerEG	VariationalEG
Training Time	2h30mins	20mins	25mins
Inference Time	10mins	30s	33s

D. Efficiency of VariationalEG

Since the VariationalEG involves sampling latent variable $\mathbf{z}$, we investigate the time consumption of different explanation generators. Table V shows the efficiency of RNN-based explanation generator and Transformer-based explanation generators. RNN-based generator ATTENTION consumes about 2h30mins to train and 10mins to inference per epoch. Thanks to the parallel computation in Transformer, time consumption of training TransformerEG decreases to about 20mins per epoch on the same hardware, gaining 7.5x speedup compared to RNN-based generator. Inference time of TransformerEG decreases to about 30s per epoch and gains 20x speedup. In VariationalEG, we perform one sampling per data point on the latent variable. Time consumption of training VariationalEG increases to about 25mins per epoch,

requiring more 25% time. While evaluating VariationalEG consumes about more 10% time per epoch compared to TransformerEG. However, the speed of VariationalEG is still much faster than the RNN-based generator. To summarize, the proposed VariationalEG does not only achieve better performance compared to the existing explanation generators, also consumes less time to train and inference.

E. Implementation Details

We implement the proposed VariationalEG using PyTorch [22] and train it on Tesla v100. We use Adam optimizer [23] with an initial learning rate of 0.0001 and minimum learning rate of 10^{-5}. If the loss on development set does not decrease compared to the previous epoch, learning rate will be decayed in half. Batch size is set as 64. We apply Dropout [24] in Transformer with dropout rate = 0.4 to avoid overfitting. We replace words whose frequency less than 15 with unknown symbol $\langle$UNK$\rangle$. Embeddings of TransformerEG and VariationalEG are randomly initialized with Xavier [25]. Transformers in encoder and decoder are set as 512 dimensions, 8 heads 6 layers, and 1024 inner dimensions. We set the dimension of latent variable $\mathbf{z}$ as 300. To avoid posterior collapse in VariationalEG, we use logistic function as KL-Annealing [7] function within the first 15000 batches. Additionally, the weight of logic supervision α is set as 1.0.

V. Conclusion

In this paper, we propose a novel generative explanation generator for NLI called Variational Explanation Generator (VariationalEG). VariationalEG introduces a latent variable to model the global logic information between premise-hypothesis pair and explanation. Additionally, a simple yet effective method called Logic Supervision is introduced to avoid the posterior collapse in VariationalEG. Experiments on the benchmark of explanation generation of NLI show that the proposed VariationalEG achieves significant improvement compared to the base model and yields a new state-of-the-art result on e-SNLI. In future work, we hope to make latent variable more interpretable and enhance the influence of latent variable on decoder.

Acknowledgment

We would like to thank the anonymous reviewers for their insightful comments. This work is supported by the National Key R&D Program of China (No.2018YFB1005102) and the NSFC (No.61976114 and No.61936012).

References

[1] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," in *BlackboxNLP@EMNLP*, 2018.
[2] O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom, "e-snli: Natural language inference with natural language explanations," in *NeurIPS*, 2018.
[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014.
[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.
[5] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015.
[6] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *NIPS*, 2015.
[7] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating sentences from a continuous space," in *CoNLL*, 2015.
[8] S. Zhao, J. Song, and S. Ermon, "Infovae: Information maximizing variational autoencoders," *ArXiv*, vol. abs/1706.02262, 2017.
[9] T. Zhao, R. Zhao, and M. Eskénazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," in *ACL*, 2017.
[10] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.
[11] B. MacCartney and C. D. Manning, *Natural language inference*. Citeseer, 2009.
[12] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *ICML*, 2014.
[13] H. Shah and D. Barber, "Generative neural machine translation," in *NeurIPS*, 2018.
[14] I. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *AAAI*, 2016.
[15] T. Zhao, K. Lee, and M. Eskénazi, "Unsupervised discrete sentence representation learning for interpretable neural dialog generation," in *ACL*, 2018.
[16] Y. Bao, H. Zhou, S. Huang, L. Li, L. Mou, O. Vechtomova, X. Dai, and J. Chen, "Generating sentences from disentangled syntactic and semantic spaces," in *ACL*, 2019.
[17] B. Zhang, D. Xiong, J. Su, H. Duan, and M. Zhang, "Variational neural machine translation," in *EMNLP*, 2016.
[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2018.
[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.
[20] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *EMNLP*, 2015.
[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2001.
[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. D.-I. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS 2019*, 2019.
[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.
[24] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.