MGR: Multi-generator Based Rationalization

Wei Liu[1]   **Haozhao Wang**[1][*]   **Jun Wang**[2][†]   **Ruixuan Li**[1]
Xinyang Li[1]   **Yuankai Zhang**[1]   **Yang Qiu**[1]
[1]School of Computer Science and Technology,
[1]Huazhong University of Science and Technology, Wuhan City, Hubei Province, China
[2]iWudao Tech
[1]{idc_lw, hz_wang, rxli, lxy722, yuankai_zhang, anders}@hust.edu.cn
[2]jwang@iwudao.tech

Abstract

Rationalization is to employ a generator and a predictor to construct a self-explaining NLP model in which the generator selects a subset of human-intelligible pieces of the input text to the following predictor. However, rationalization suffers from two key challenges, i.e., spurious correlation and degeneration, where the predictor overfits the spurious or meaningless pieces solely selected by the not-yet well-trained generator and in turn deteriorates the generator. Although many studies have been proposed to address the two challenges, they are usually designed separately and do not take both of them into account. In this paper, we propose a simple yet effective method named MGR to simultaneously solve the two problems. The key idea of MGR is to employ multiple generators such that the occurrence stability of real pieces is improved and more meaningful pieces are delivered to the predictor. Empirically[1], we show that MGR improves the F1 score by up to 20.9% as compared to state-of-the-art methods.

1 Introduction

The widespread use of deep learning in NLP models has led to increased concerns about interpretability. To solve this problem, Lei et al. (2016) proposed rationalization framework RNP in which a generator selects human-intelligible subsets (i.e., rationales) from the input text and feeds them to the subsequent predictor that maximizes the text classification accuracy, as shown in Figure 1. Unlike post-hoc approaches for explaining black-box models, the RNP framework has the built-in self-explaining ability through a cooperative game between the generator and the predictor. RNP and its variants have become one of the mainstreams

---
[*]Corresponding author
[†] This paper is a collaborative work between the Intelligent and Distributed Computing Laboratory at Huazhong University of Science and Technology, and iWudao Tech.
[1]https://github.com/jugechengzi/Rationalization-MGR

Figure 1: The standard rationalization framework RNP. $X, Z, \hat{Y}, Y$ represent the input text, rationale, prediction and the groundtruth label, respectively.

to facilitate the interpretability of NLP models (Yu et al., 2021; Liu et al., 2022, 2023). Notably, given the versatility of the self-explaining rationalization framework, such methods have significant potential for application in diverse fields such as multi-aspect recommender systems (Deng et al., 2023) and computer vision (Yuan et al., 2022).

Despite its strength, rationalization schemes are notoriously hard to train. Two main training obstacles are the spurious correlations (Chang et al., 2020) and the degeneration (Yu et al., 2019). As shown in the example of Table 1(a), the problem of spurious correlations is that the predictor mistakenly makes a correlation between the label on some specific aspect and the spurious pieces on another similar aspect, which commonly exists in multi-aspect classification (Chang et al., 2020; Plyler et al., 2021; Yue et al., 2023). Degeneration means that the predictor may overfit to meaningless rationales generated by the not yet well-trained generator (Yu et al., 2019), causing the converged generator tends to select these uninformative rationales, which is illustrated in the example of Table 1(b).

Many prior efforts have separately considered the problem of spurious correlations or degeneration in rationalization. For instance, to solve the problem of spurious correlations, some recent methods leverage the idea of causal inference to build the causal relationship between the rationale and label (Chang et al., 2020; Yue et al., 2023). The common idea to address the degeneration problem is to introduce some auxiliary modules such that the predictor has access to the full texts, and thus

it cannot overfit the meaningless rationales solely provided by the generator (Yu et al., 2019; Huang et al., 2021; Yu et al., 2021).

Although these approaches may be effective at either solving the problem of spurious correlations or degeneration isolation, they are usually designed separately and do not take both of them into account. In this paper, we seek to simultaneously solve the two problems. Specifically, we identify that both two problems arise from that the predictor has only access to the limited view of pieces provided by the single generator, and thus may learn corruptly when this generator selects spurious or meaningless rationales. Besides, recent studies find that the initialization of the model has a significant impact over the training performance, which implicitly indicates that the rationalization model is hardly to train once the single generator is not well initialized (Jain et al., 2020; Yu et al., 2021).

Considering these limitations of the rationalization with one single generator, as shown in Figure 2, we design a novel architecture where there is a predictor but with multiple generators. These generators are initialized with different parameters. In this way, the view of the predictor is not limited to one single generator and it can have access to more meaningful rationales. We theoretically show that the occurrence stability of real rationales increases such that the predictor has lower risks at learning spurious correlations, and that the diversity of the rationales is improved such that the predictor can hardly deviate to some specific meaningless rationale. Extensive experiments conducted on three widely used rationalization benchmarks, i.e., the correlated BeerAdvocate dataset (McAuley et al., 2012), the decorrelated BeerAdvocate dataset (Lei et al., 2016), and the Hotel Reviews dataset (Wang et al., 2010), show that MGR achieves significant improvements over several state-of-the-art methods in terms of the rationale quality. Our contributions can be summarized as:

• To the best of our knowledge, this paper is the first to simultaneously solve the spurious correlations and degeneration problem in rationalization. We propose a simple but effective method, namely, MGR, that facilitates the predictor to have a broader view of the rationales by using multiple generators.

• We theoretically prove that using multiple generators can provide real rationales more stably such that the risk of the predictor learning spurious correlations is reduced. Besides, we prove that multiple generators can produce more diverse rationales and thus the predictor will not overfit to some specific meaningless rationale.

• We conduct extensive experiments over various datasets and show that MGR achieves an improvement by up to 20.9% as compared to state-of-the-art rationalization methods in terms of F1 score.

2 Related Work

2.1 Rationalization

The base cooperative framework of rationalization named RNP (Lei et al., 2016) is flexible and offers a unique advantage, i.e., certification of exclusion, which means any unselected input is guaranteed to have no contribution to prediction (Yu et al., 2021). However, such a method is hard to train. To tackle this challenge, many methods have been proposed to improve RNP from different aspects.

Rationales sampling. Many works focus on refining the sampling process of the rationales. Bao et al. (2018) used Gumbel-softmax to do the reparameterization for binarized selection. Bastings et al. (2019) replaced the Bernoulli sampling distributions with rectified Kumaraswamy distributions. Jain et al. (2020) disconnected the training regimes of the generator and predictor via a saliency threshold. Paranjape et al. (2020) imposed a discrete bottleneck objective to balance the task performance and the rationale length. Hase et al. (2020) explored better metrics for the explanations. Rajagopal et al. (2021) used phrase-based concepts to conduct a self-explaining model. These methods are orthogonal to our method.

Degeneration. Degeneration is one of the major problem in rationalization. To solve this problem, many efforts seek to regularize the predictor using supplementary modules which have access to the information of the full text (Yu et al., 2019; Huang et al., 2021; Yu et al., 2021) such that the generator and the predictor will not collude to uninformative rationales. 3PLAYER (Yu et al., 2019) takes the unselected text Z^c into consideration by inputting it to a supplementary predictor $Predictor^c$. DMR (Huang et al., 2021) tries to align the distributions of rationale with the full input text in both the output space and feature space. A2R (Yu et al., 2021) endows the predictor with the information of full text by introducing a soft rationale.

Spurious correlations. The predictor in rationalization model may make correlations between spurious rationales and the label. To tackle this chal-

Label (Aroma): Positive   **Prediction:** Positive
Text: the appearance was nice . dark gold with not much of a head but nice lacing when it started to dissipate . <u>the smell was ever so hoppy with a hint of the grapefruit flavor that 's contained within . </u> the taste was interesting , up front tart grapefruit , not sweet in the least . more like grapefruit rind even . slight hint of hops and seemingly no malt . the mouth feel was crisp , with some biting carbonation . drinkability was easily above average due to the crispness and lack of sweetness . not the usual taste you expect when drinking a fruit beer . in fact this is my favorite fruit beer ever .

(b) An example of degeneration

Label (Appearance): Negative   **Prediction:** Negative
Text: <u>appearance : light yellow to almost clear</u> smell : slight hops , but barely smelled like beer taste : little to none , like a rice lager , zero taste mouthfeel : watery and tasteless drinkability : very easy , goes down easier than water . good for drinking games.

Table 1: The blue piece of the text is the human-annotated rationale. Pieces of the text with <u>underline</u> are the rationales from RNP. (a): An example of RNP making the right sentiment prediction using the spurious correlation. If the predictor overfits the spurious correlation, it will then tell the generator to continue to select this spurious correlation as the rationale. (b): An example of RNP making the right sentiment prediction using an uninformative rationale. Initially, the generator may randomly select some uninformative candidates like "appearance" as rationales for the negative text. The predictor of RNP overfits to these uninformative rationales and classifies the sentiment according to whether "appearance" is included in the rationale. Guided by such a spoiled predictor, the generator in turn tends to select these uninformative rationales.

lenge, the typical methods mainly adopt causal inference to exclude the spurious correlations. For instance, Chang et al. (2020) introduced an environment-agnostic predictor to recognize the spurious correlations. Yue et al. (2023) aimed to remove the spurious correlations based on the back-door adjustment.

2.2 Model Ensemble Methods

Ensemble methods that combine the outputs of several different models to improve the prediction performance and robustness have been studied for a long time (Breiman, 1996; Wolpert, 1992; Schapire, 1999). Ensemble methods train N models with different datasets independently and fuse the outputs of different models during inference. This requires maintaining N models and running each of them at test time, which increases the costs of computational resources and brings obstacles to applications (Singh and Jaggi, 2020). Although our method is similar to ensemble methods to some extent, it has essential differences with ensemble methods. In our method, different generators are not trained entirely independently. In fact, they all play a cooperative game with the same predictor on one same training dataset. With different initializations, different generators can provide diverse

rationales to train a robust and stable predictor at early training stage. But with the same training target and dataset, different generators can converge to get the same output (Ainsworth et al., 2022), thus we only need to keep one generator during inference, which is also empirically supported by the experimental results in Figure 3(b).

3 Problem Definition

Notation. $f_G(\cdot)$ and $f_P(\cdot)$ represent the generator and predictor, respectively. θ_G and θ_P represent the parameters of the generator and predictor, respectively. $\mathcal{D}$ represents the distribution of dataset. We consider the classification problem, where the input is a text sequence $X=[x_1, x_2, \cdots, x_l]$ with x_i being the i-th token and l being the number of tokens. The label of X is a one-hot vector $Y \in \{0,1\}^c$, where c is the number of categories.

Cooperative rationalization. Rationalization framework consists of a generator and a predictor. The goal of the generator is to find the most informative pieces containing several tokens in the original input text X. For each sample $(X, Y) \sim \mathcal{D}$, the generator firstly outputs a sequence of binary mask $M = [m_1, \cdots, m_l] \in \{0, 1\}^l$. Then, it forms the rationale Z by the element-wise product of X
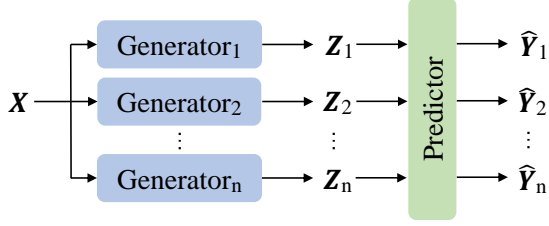
Figure 2: The architecture of MGR. X is the input full text. Z_i is the selected rationale and $\hat{Y}_i$ is the predictor's output corresponding to Z_i.

and M:

$$Z = M \odot X = [m_1 x_1, \cdots, m_l x_l]. \quad (1)$$

To simplify the notation, we denote $Z = f_G(X)$. In cooperative rationalization, the informativeness of the rationale Z provided by the generator is measured by the negative cross entropy $-H(Y, \hat{Y}_z)$, where $\hat{Y}_z$ is the output of the predictor with the input being Z. Consequently, the generator and the predictor are usually optimized cooperatively:

$$\min_{\theta_G, \theta_P} \sum_{(X,Y) \sim \mathcal{D}} H(Y, f_P(f_G(X))). \quad (2)$$

Regularizer of shortness and coherence. To make the selected rationales human-intelligible, the original RNP constrains the rationales by short and coherent regularization terms. In this paper, we use the constraints updated by Chang et al. (2019):

$$\Omega(M) = \lambda_1 \left| \frac{\|M\|_1}{l} - s \right| + \lambda_2 \sum_{t=2}^{l} |m_t - m_{t-1}|. \quad (3)$$

The first term encourages that the percentage of the tokens being selected as rationales is close to a pre-defined level s. The second term encourages the rationales to be coherent.

4 Multi-generator based Rationalization

4.1 Methodology

Based on the framework of RNP, MGR uses multiple generators with different initialized parameters to help the predictor learn from diverse rationale candidates, as shown in Figure 2. For the convenience of comparing with previous methods in experiments, we adopt the bidirectional gated recurrent units (GRUs) (Cho et al., 2014) as the encoder which has been adopted by most previous works (Chang et al., 2020; Huang et al., 2021; Yu et al., 2021).

Training of MGR. MGR is to leverage each generator $f_{G_i}(\cdot)$ to process the input text X in isolation, and then send the obtained rationales Z_i to the predictor $f_P(\cdot)$ to obtain the prediction $\hat{Y}_i$. Based on Equation 2, MGR computes its loss by calculating the sum of the cross entropy between Y and $\hat{Y}_i$ in each generator:

$$\min_{\theta_{G_1}, \dots, \theta_{G_n}, \theta_P} \sum_{(X,Y) \sim \mathcal{D}} H(Y, \hat{Y})$$

$$= \sum_{(X,Y) \sim \mathcal{D}} \sum_{i=1}^{n} H(Y, f_P(f_{G_i}(X))). \quad (4)$$

Inference of MGR. During inference, MGR only uses one generator, e.g., the first generator Z_1 and $\hat{Y}_1$, to provide the rationale. It's worth noting that our idea is similar to but not the same as ensemble models. We keep a set of *generators* during training to help train a good predictor which in turn will promise the cooperative generator to be good. Nevertheless, we only keep the first *generator* during inference (see experimental results in Figure 3(b)), which is efficient in terms of both time and resource consumption.

4.2 Dealing with spurious correlations

In this section, we seek to show that the proposed method can solve the problem of spurious correlations. Specifically, we specify the principles of our method on the case with simplified settings.

Settings. Similar to Yu et al. (2021), we consider the input text X consists of two subsets of the features X_1 and X_2, where X_1 is the causal rationale belonging to the target aspect label, while X_2 is the comments about other aspects that correlates with the target aspect. We denote the parameters of the predictor by θ_P^1 when it fits X_1, and by θ_P^0 when it fits X_2. We also assume a payoff table as shown in Table 2, where we denote the negative loss as the payoff. The higher payoff indicates that the rationalization achieves better convergence results. Considering that rationalization is a cooperative game, the payoff of the generator is the same as that of the predictor. Here, a can be seen as the negative loss when the predictor fits the generator's selection and b is the negative loss when the predictor fits the unselected part. Since the loss is usually smaller when $f_P(\cdot)$ fits what $f_{G_i}(\cdot)$ selects, we consider $a > b$. We denote that there are k *generators* that select the causal rationale X_1 and the other $n - k$ *generators* select X_2. Without losing generality, we consider that the predictor

$f_P(\cdot)$ / $f_{G_i}(\cdot)$	θ_P^1	θ_P^α	θ_P^0
Select X_1	(a,a)	$(\alpha \cdot a + (1-\alpha) \cdot b, \alpha \cdot a + (1-\alpha) \cdot b)$	(b,b)
Select X_2	(b,b)	$((1-\alpha) \cdot a + \alpha \cdot b, (1-\alpha) \cdot a + \alpha \cdot b)$	(a,a)

Table 2: The payoff (negative loss) table of the cooperative game between the generator and the predictor. Here $a > b$, which indicates that if $f_P(\cdot)$ fits what $f_{G_i}(\cdot)$ selects, both $f_P(\cdot)$ and $f_{G_i}(\cdot)$ get a higher payoff.

is randomly initialized by interpolation between θ_P^0, θ_P^1 with θ_P^α, where α is a random variable in [0,1]. Similar to Yu et al. (2021)[2], we consider α to be the probability (or degree) that the predictor tends to fit X_1. If $\alpha = 1$, the predictor always fits X_1, and if $\alpha = 0$, the predictor always fits X_2. This is where the third column in Table 2 comes from.

Theoretical analysis. For the situation that k *generators* select X_1 and the predictor is θ_P^α, the expected payoff of the predictor is

$$R_P(\alpha) = k \cdot \alpha \cdot a + k(1-\alpha) \cdot b \\ + (n-k)(1-\alpha)a + (n-k) \cdot \alpha \cdot b. \quad (5)$$

With the generator's random initialization, k follows a binomial distribution $\mathcal{B}(n, P_c)$, where P_c depends on the dataset (subscript c stands for causality). We assume that $P_c > 0.5$ because the causality appears more often than the spurious correlation when the target aspect's label appears (Please refer to Appendix B.4 and Table 11 for more detailed discussion). Apparently, we have

$$\lim_{n \to \infty} p(k < n - k) = 0, \quad (6)$$

of which the detailed derivation is in Appendix B.1.

Lemma 1 *If the number of generators that select X_1 is more than those that select X_2 (i.e., $k > n-k$), the predictor will be optimized to increase α to get a higher payoff (i.e., $\frac{\partial R_P(\alpha)}{\partial \alpha} > 0$).*

The proof is in Appendix B.2. Lemma 1 indicates that if $k > n - k$, by increasing $R_P(\alpha)$ (for lower loss), the cooperative game will guide the predictor to move towards θ_P^1 (by increasing α) and fit the right causal rationale X_1. In turn, such a good predictor will guide the generator to select the desired causal rationales.

We denote the probability of the predictor overfits to the spurious correlation as $p_{MGR}(spu)$. Ac-

cording to Lemma 1, we have

$$p_{MGR}(spu) = p\left(\frac{\partial R_P(\alpha)}{\partial \alpha} < 0\right) \\ = p(k < n - k) \quad (7) \\ = \sum_{k=0}^{(n-1)/2} \left(\binom{n}{k} \cdot P_c^k \cdot (1-P_c)^{n-k}\right),$$

where $\binom{n}{k}$ is the combinatorial number. Combining Lemma 1 with Equation 6, we have the following theorem:

Theorem 1 *For any tolerable upper bound probability P_s of the predictor overfitting to spurious correlation, if $P_c > 0.5$, there always exists a constant N such that*

$$\forall n > N, \quad p_{MGR}(spu) < P_s. \quad (8)$$

The proof is deferred to Appendix B.3. Theorem 1 indicates that we can reduce the risk of spurious correlations to arbitrarily low levels by increasing the number of the generator. When $n = 1$, MGR becomes the vanilla RNP and we have

$$p_{RNP}(spu) = 1 - P_c. \quad (9)$$

It is obvious that

$$\forall n > 1, \quad p_{MGR}(spu) < p_{RNP}(spu). \quad (10)$$

4.3 Dealing with Degeneration

In this section, we consider X, Y, Z as random variables rather than deterministic ones. The principle behind Equation 2 is to maximize the mutual information between Y and Z (Chang et al., 2020; Yu et al., 2021):

$$\max_Z I(Y; Z) = \max_Z (H(Y) - H(Y|Z)). \quad (11)$$

Since $H(Y)$ is irrelevant to Z, the equation is equal to minimizing $H(Y|Z)$. Degeneration happens because the diversity of the rationales is not taken into account. Hence, the generator of RNP may get rationale candidates with low diversity (i.e., low $H(Z)$). Under this case, the predictor may overfit to some specific patterns that are contained in the limited rationale candidates and has a high risk of occurring degeneration when the rationales are merely noises.

Next, we show that MGR with multiple generators improves the diversity of the selected rationale candidates. Specifically, by viewing the rationales of different generators as different variables, we can compute the rationale diversity of MGR as

$$H(Z_{MGR}) = H(Z_1, Z_2, \cdots, Z_n). \quad (12)$$

---

[2] Our α is similar to π in (Yu et al., 2021). In (Yu et al., 2021), π is the degree that the generator tends to select X_1.

Theorem 2 *For $\forall i \in [1, n]$, we have*

$$H(Z_i) \leq H(Z_1, Z_2, \cdots, Z_n) \leq \sum_{k=1}^{n} H(Z_k), \quad (13)$$

where the right equality holds if and only if $\forall i, j, \ Z_i \perp\!\!\!\perp Z_j$, and the left equality holds if and only if $\forall i, j, \ Z_i = Z_j$.

The proof is in Appendix B.5. Theorem 2 indicates that the diversity of MGR with multiple generators is equivalent to the case with one single generator when all generators are the same. More specifically, since RNP consists of only one generator, we have

$$H(Z_{RNP}) = H(Z_i) \leq \max_i H(Z_i), \quad (14)$$

where $i \in [1, n]$. We always have $H(Z_{MGR}) \geq H(Z_{RNP})$ no matter how different generators are coupled together, thus alleviating degeneration.

Besides, Theorem 2 also indicates that the diversity of MGR achieves the maxima when all the generators are completely independent. Accordingly, we seek to decrease the correlation between different generators to make that $H(Z_{MGR})$ gets closer to $\sum_{k=1}^{n} H(Z_k)$ during training which is specified in the next section.

4.4 Diverse Training with Separate Learning Rates

To facilitate the improvement of the diversity of rationales while guaranteeing the convergence of rationalization models, we consider that training MGR has to satisfy two conditions. First, to deal with degeneration, generators should be different from each other to guarantee that the predictor continuously learns from diverse rationales before it learns adequate information. Second, different generators should be able to achieve the same convergence result, i.e., selecting the same rationales for any given text, after the predictor has learned enough information and converged. Only in this way can we keep one single generator during inference to guarantee that MGR is efficient in terms of latency and resource consumption.

To satisfy the two properties, we propose separately setting the learning rates of different generators. Intuitively, separate learning rates provide different generators with different learning states in any training moment, thus keeping them diverse during the learning process. On the other side, learning rates do not modify the loss landscape of generators and thus these generators can eventually achieve the same convergence result although
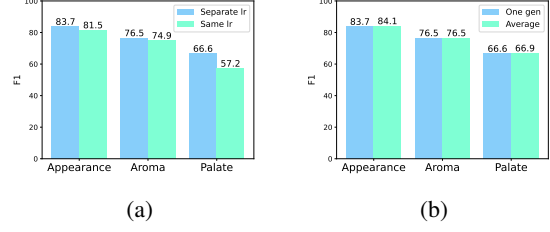

(a)   (b)

Figure 3: Experiments on *decorrelated BeerAdvocate* verifying (a) the effectiveness of separate learning rates and (b) the feasibility of keeping only the first generator.

maybe at different speeds (Ainsworth et al., 2022). The argument is also empirically supported by the results in Figure 3(b).

Formally, we denote the learning rate of the i-th generator as η_i and the loss as $\mathcal{L}$. *generator$_i$* and *generator$_j$* are updated during training as:

$$\begin{aligned} \theta_{G_i}' &= \theta_{G_i} - \eta_i \cdot \nabla_{\theta_{G_i}} \mathcal{L}, \\ \theta_{G_j}' &= \theta_{G_j} - \eta_j \cdot \nabla_{\theta_{G_j}} \mathcal{L}, \end{aligned} \quad (15)$$

In practice, we first find a learning rate η and set the i-th generator's learning rate simply to be $i \cdot \eta$. And to alleviate the problem that the loss function of the predictor is too large due to the superposition of multiple cross-entropies from different generators, we set the learning rate of the predictor to be $\frac{\eta}{n}$.

To support our claims, we conduct two practical experiments on *decorrelated BeerAdvocate* dataset, where the main problem is degeneration. First, we compare the performance of MGR using one learning rate to MGR using separate learning rates. The results are shown in Figure 3(a). Although using separate learning rates does not help much in the relatively easy aspects including *Appearance* and *Aroma*, it makes a significant improvement in the hard *Palate* aspect. Second, we compare the performance of keeping only one *generator* for inference to averaging the results of multiple *generators*, as shown in Figure 3(b). The results show that keeping only one *generator* hardly influences the performance, which indicates that different *generators* can finally converge to get the same outputs and only one *generator* is required in inference time. We also show the differences in the rationales generated by different generators in Figure 7 of Appendix A.5.

Methods	Appearance					Aroma					Palate				
	S	Acc	P	R	F1	S	Acc	P	R	F1	S	Acc	P	R	F1
RNP*	10.0	-	32.2	18.6	23.6	10.0	-	44.8	32.4	37.6	10.0	-	24.6	23.5	24.0
INVRAT*	10.0	-	42.6	31.5	36.2	10.0	-	41.2	39.1	40.1	10.0	-	34.9	45.6	39.5
Inter-RAT*	11.7	-	66.0	46.5	54.6	11.7	-	55.4	47.5	51.1	12.6	-	34.6	48.2	40.2
MGR(ours)	10.9	80.5	**87.5**	**51.7**	**65.0**	10.3	89.7	**78.7**	**52.2**	**62.8**	10.8	86.0	**65.6**	**57.1**	**61.1**
RNP*	20.0	-	39.4	44.9	42.0	20.0	-	37.5	51.9	43.5	20.0	-	21.6	38.9	27.8
INVRAT*	20.0	-	58.9	67.2	62.8	20.0	-	29.3	52.1	37.5	20.0	-	24.0	55.2	33.5
Inter-RAT*	21.7	-	62.0	76.7	68.6	20.4	-	44.2	65.4	52.8	20.8	-	26.3	59.1	36.4
MGR(ours)	20.3	85.6	**76.3**	**83.6**	**79.8**	19.7	89.6	**64.4**	**81.3**	**71.9**	19.3	89.3	**47.1**	**73.1**	**57.3**
RNP*	30.0	-	24.2	41.2	30.5	30.0	-	27.1	55.7	36.4	30.0	-	15.4	42.2	22.6
INVRAT*	30.0	-	41.5	74.8	53.4	30.0	-	22.8	65.1	33.8	30.0	-	20.9	**71.6**	32.3
Inter-RAT*	30.5	-	48.1	82.7	60.8	29.4	-	37.9	72.0	49.6	30.4	-	21.8	66.1	32.8
MGR(ours)	30.4	88.5	**57.2**	**93.9**	**71.1**	29.8	91.6	**45.8**	**87.4**	**60.1**	30.3	89.3	**27.3**	66.5	**38.7**

Table 3: Results on *correlated BeerAdvocate*. " ∗ ": results obtained from Inter-RAT (Yue et al., 2023).

	RNP	CAR	DMR	A2R	MGR(Ours)
modules	1gen+1pred	1gen+2pred	1gen+3pred	1gen+2pred	3gen+1pred
parameters	2×	3×	4×	3×	4×

Table 4: The complexity of different models. "gen": generator. "pred": predictor.

5 Experiments

5.1 Experimental Setup

Datasets 1) **BeerAdvocate** (McAuley et al., 2012) is a multi-aspect sentiment prediction dataset widely used in rationalization. There is a high correlation among the rating scores of different aspects in the same review, making the rationale selecting faces a severe spurious correlation. We use the original dataset to verify the effectiveness of MGR in dealing with spurious correlation and degeneration at the same time. In addition, following the previous work (Lei et al., 2016; Huang et al., 2021; Yu et al., 2021), we use the subsets containing less spurious correlation (Lei et al., 2016) to see the effectiveness in dealing with solitary degeneration. 2) **Hotel Reviews** (Wang et al., 2010) is another multi-aspect sentiment classification dataset also widely used in rationalization. Each aspect itself can be seen as a dataset and is trained independently.

Baselines and implementation details. In parcitce, we set $n = 3$ (the number of generators) as for our MGR as a performance-time trade-off. We compare MGR to the vanilla RNP (Lei et al., 2016) and several latest published methods that achieve state-of-the-art results: INVRAT (Chang et al., 2020), DMR (Huang et al., 2021), A2R (Yu et al., 2021), Inter-RAT (Yue et al., 2023), all of which have been specified in Section 2. Following the commonly used rationalization settings (Chang et al., 2019; Yu et al., 2019; Chang et al., 2020; Huang et al., 2021; Yu et al., 2021; Yue et al., 2023), we use the 100-dimension Glove (Pennington et al., 2014) as the word embedding and 200-dimension GRUs to get the text representation. We do not use BERT (Devlin et al., 2019) because it is still a challenging task to finetune large pretrained models on the RNP cooperative framework (see Table 4 in Chen et al. (2022) and Appendix A.2). We use Adam (Kingma and Ba, 2015) as the optimizer. All the baselines are tuned multiple times manually to find the best hyperparameters. The complexity of different models are shown in Table 4 All of the models are trained on a RTX3090 GPU. More details are in Appendix A.1.

Metrics. All the methods get similar predictive accuracy. Following (Chang et al., 2020; Huang et al., 2021; Yu et al., 2021; Yue et al., 2023), we mainly focus on the quality of rationales, which is measured by the overlap between the model-selected tokens and human-annotated tokens. $P, R, F1$ indicate the precision, recall, and F1 score, respectively. S indicates the average sparsity of the selected rationales, i.e., the percentage of selected tokens to the whole texts. Acc indicates the predictive accuracy of the test set.

5.2 Results

We first conduct an experiment on the *correlated BeerAdvocate* dataset, where the problems of degeneration and spurious correlation both may damage the rationale quality. Methods that achieve the state-of-the-art results on this dataset are INVRAT (Chang et al., 2020) and Inter-RAT (Yue et al., 2023). We tune s in Equation 3 to get similar rationale sparsity as previous methods do. The results are shown in Table 3. We improve the F1 score by up to 20.9% (*Palate* aspect with 20% sparsity) over

(a) Normal experiments on *decorrelated BeerAdvocate*

Methods	Appearance					Aroma					Palate				
	S	Acc	P	R	F1	S	Acc	P	R	F1	S	Acc	P	R	F1
RNP*	os	85.7	83.9	71.2	72.8	os	84.2	73.6	67.9	65.9	os	83.8	55.5	54.3	51.0
re-DMR	18.2	-	71.1	70.2	70.7	15.4	-	59.8	58.9	59.3	11.9	-	53.2	50.9	52.0
re-A2R	18.4	83.9	72.7	72.3	72.5	15.4	86.3	63.6	62.9	63.2	12.4	81.2	57.4	57.3	57.4
A2R*	os	86.3	84.7	71.2	72.9	os	84.9	**79.3**	71.3	70.0	os	84.0	64.2	60.9	58.0
MGR(ours)	18.4	86.1	**83.9**	**83.5**	**83.7**	15.6	86.6	76.6	**76.5**	**76.5**	12.4	85.1	**66.6**	**66.6**	**66.6**

(b) Beer-Skewed in *Palate* aspect of *decorrelated BeerAdvocate*

Setting	RNP*				A2R*				MGR(ours)			
	Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
skew10	77.3	5.6	7.4	5.5	82.8	50.3	48.0	45.5	82.0	**65.2**	**62.8**	**64.0**
skew15	77.1	1.2	2.5	1.3	80.9	30.2	29.9	27.7	77.4	**62.7**	**58.2**	**60.4**
skew20	75.6	0.4	1.4	0.6	76.7	0.4	1.6	0.6	82.5	**65.6**	**63.2**	**64.4**

Table 5: The standard experiment and one synthetic experiment on *decorrelated BeerAdvocate*. "∗": results from the paper of A2R. "re-": our reimplemented methods. "os": one sentence.

Methods	Location					Service					Cleanliness				
	S	Acc	P	R	F1	S	Acc	P	R	F1	S	Acc	P	R	F1
RNP*	10.9	-	43.3	55.5	48.6	11.0	-	40.0	38.2	39.1	10.6	-	30.5	36.0	33.0
CAR*	10.6	-	46.6	58.1	51.7	11.7	-	40.7	41.4	41.1	9.9	-	32.3	35.7	33.9
DMR**	10.7	-	47.5	60.1	53.1	11.6	-	43.0	43.6	43.3	10.3	-	31.4	36.4	33.7
re-A2R	8.5	87.5	43.1	43.2	43.1	11.4	96.5	37.3	37.2	37.2	8.9	94.5	33.2	33.3	33.3
MGR(ours)	9.7	97.5	**52.5**	**60.5**	**56.2**	11.8	96.5	**45.0**	**46.4**	**45.7**	10.5	96.5	**37.6**	**44.5**	**40.7**

Table 6: Results on *HotelReview*. Each aspect is trained independently. "∗": results from the paper of CAR (Chang et al., 2019), "∗∗": results from the paper of DMR. "re-": our reimplemented method.

the latest SOTA. Besides, except the *Palate* aspect with 30% sparsity, we get over 10% improvements under all the other settings.

We then conduct an experiment on the *decorrelated BeerAdvocate* dataset, where the main problem is degeneration. Methods that achieve the state-of-the-art results on this dataset are DMR (Huang et al., 2021) and A2R (Yu et al., 2021). Since the rationales of *BeerAdvocate* on a sentence level, A2R in its original paper does sentence level selection (i.e., selecting one sentence as the rationale) on this dataset. We also reimplement A2R according to its source codes to do the token-level selection. The results are shown in Table 4(a). The sparsity is set to be close to that of the human-annotated rationales. We beat all the methods in terms of F1 score. We do not get as significant improvements as those in Table 3 because the spurious correlation is removed manually in this dataset. But we still get up to 10.8% (*Appearance* aspect) improvements as compared to the SOTA.

To show the generalizability of our method, we further conduct an experiment on *HotelReviews*. Methods that achieve the state-of-the-art results on this dataset are DMR (Huang et al., 2021) and

Methods	Appearance			
	S	Acc	P	R	F1
MGR(n=5)	19.2	86.3	83.8	86.8	85.3
MGR(n=7)	19.6	87.0	83.5	88.3	85.8
MGR(n=9)	19.4	86.0	83.6	87.7	85.6

Table 7: Results of MGR with different numbers of generators. The dataset is the *Appearance* aspect of *correlated BeerAdvocate*. The sparsity is $S \approx 20$.

CAR (Chang et al., 2019). We also beat all the baselines and get up to 6.8% (*Cleanliness* aspect) improvements on this dataset.

Results of MGR with different numbers of generators. Although we set $n = 3$ in our previous experiments, we also show the results of our MGR with different values of n in Table 7. When n grows, the results are somewhat better than those of $n = 3$. However, $n = 3$ yields the most improvements per additional generator and proved to be a good performance-cost trade-off. And note that, having too many generators may not always result in better outcomes, because the learning rate for the i-th generator, which is $i \times \eta$, may become too large for stable training.
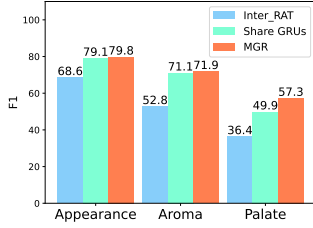
Figure 4: The comparison of sharing the encoders (GRUs) of MGR's generator.

Beer-Skewed. To show that our MGR does not suffer from the degeneration problem, we conduct the same synthetic experiment that deliberately induces degeneration as Yu et al. (2021) did. The details of the experimental setup are in Appendix A.3. We use the relatively harder *Palate* aspect (Yu et al., 2021). The results are shown in Table 4(b). The results of RNP and A2R are obtained from (Yu et al., 2021). For all the settings, we outperform both RNP and A2R. Especially, for *skew*20, RNP and A2R can not work at all while our MGR is only slightly influenced as compared to the corresponding result in Table 4(a).

Sharing encoders between generators. The major limitation of MGR is the increased computational costs. One plausible trade-off method may be sharing some parameters between the generators. We conduct an experiment where we share the generators' GRUs but keep their own linear heads. Figure 4 shows the results on *correlated BeerAdvocate* with sparsity around 20%. More results are in Appendix A.4. Although simply sharing the generators' encoders sometimes cause damage to the performance of MGR, it still outperforms the state-of-the-art method Inter_RAT. We leave how to better decrease the computational costs without hurting the model performance as future work.

6 Conclusion and future work

In this paper, we design a new framework MGR to simultaneously tackle the two major challenges including the spurious correlation and degeneration in the self-explaining rationalization schemes. Specifically, we propose leveraging multiple generators to select rationales such that the predictor can have access to more meaningful rationales stably. We theoretically show that the proposed method can solve the two problems. Finally, empirical results conducted on various datasets demonstrate the great effectiveness of our proposed method.

Limitations

More generators bring significant benefits to the model performance to our MGR, but the training cost is also increased with the number of generators growing. Although we have verified that we only need to keep one generator during test, there is no denying that the training cost is still an important problem. In the future, we will explore some methods like multi-task learning and model fusion, to reduce the model complexity.

Acknowledgements

References

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha S. Srinivasa. 2022. Git re-basin: Merging models modulo permutation symmetries. *CoRR*, abs/2209.04836.

Yujia Bao, Shiyu Chang, Mo Yu, and Regina Barzilay. 2018. Deriving machine attention from human rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1903–1913. Association for Computational Linguistics.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2963–2977. Association for Computational Linguistics.

Leo Breiman. 1996. Bagging predictors. *Mach. Learn.*, 24(2):123–140.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi S. Jaakkola. 2019. A game theoretic approach to classwise selective rationalization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 10055–10065.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi S. Jaakkola. 2020. Invariant rationalization. In *Proceedings of the 37th International Conference on*

Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, pages 1448–1458. PMLR.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. 2022. Can rationalization improve robustness? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3792–3805. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Zhiying Deng, Jianjun Li, Zhiqiang Guo, and Guohui Li. 2023. Multi-aspect interest neighbor-augmented network for next-basket recommendation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Nuno Miguel Guerreiro and André F. T. Martins. 2021. SPECTRA: sparse structured text rationalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6534–6550. Association for Computational Linguistics.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367.

Yongfeng Huang, Yujun Chen, Yulun Du, and Zhilin Yang. 2021. Distribution matching for rationalization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13090–13097. AAAI Press.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. 2020. Learning to faithfully rationalize by construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4459–4473. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 107–117. The Association for Computational Linguistics.

Wei Liu, Haozhao Wang, Jun Wang, Ruixuan Li, Chao Yue, and YuanKai Zhang. 2022. Fr: Folded rationalization with a unified encoder. In *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc.

Wei Liu, Jun Wang, Haozhao Wang, Ruixuan Li, Yang Qiu, Yuankai Zhang, Jie Han, and Yixiong Zou. 2023. Decoupled rationalization with asymmetric learning rates: A flexible lipschitz restraint. *CoRR*, abs/2305.13599.

Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 1020–1025. IEEE Computer Society.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1938–1952. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Mitchell Plyler, Michael Green, and Min Chi. 2021. Making a (counterfactual) difference one rationale at a time. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28701–28713.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard H Hovy, and Yulia Tsvetkov. 2021. SELFEXPLAIN: A self-explaining architecture for neural text classifiers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 836–850, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robert E. Schapire. 1999. A brief introduction to boosting. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 1401–1406. Morgan Kaufmann.

Sidak Pal Singh and Martin Jaggi. 2020. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 783–792. ACM.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–259.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi S. Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. 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, Hong Kong, China, November 3-7, 2019*, pages 4092–4101. Association for Computational Linguistics.

Mo Yu, Yang Zhang, Shiyu Chang, and Tommi S. Jaakkola. 2021. Understanding interlocking dynamics of cooperative rationalization. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12822–12835.

Hao Yuan, Lei Cai, Xia Hu, Jie Wang, and Shuiwang Ji. 2022. Interpreting image classifiers by generating discrete masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2019–2030.

Linan Yue, Qi Liu, Li Wang, Yanqing An, Yichao Du, and Zhenya Huang. 2023. Interventional rationalization.

Datasets		Train		Dev		Annotation		
		Pos	Neg	Pos	Neg	Pos	Neg	Sparsity
Beer	Appearance	202385	12897	28488	1318	923	13	18.5
	Aroma	172299	30564	24494	3396	848	29	15.6
	Palate	176038	27639	24837	3203	785	20	12.4
Beer*	Appearance	16891	16891	6628	2103	923	13	18.5
	Aroma	15169	15169	6579	2218	848	29	15.6
	Palate	13652	13652	6740	2000	785	20	12.4
Hotel	Location	7236	7236	906	906	104	96	8.5
	Service	50742	50742	6344	6344	101	99	11.5
	Cleanliness	75049	75049	9382	9382	99	101	8.9

Table 8: Statistics of datasets used in this paper. *: the decorrelated BeerAdvocate.

A   More Results

A.1   More implementation details

To the best of our knowledge, both datasets are sufficiently anonymized to make identification of individuals impossible without significant effort. Both datasets are in English. For *correlated Beer-Advocate*, we preprocess the data in the same way as Yue et al. (2023). For *decorrelated BeerAdvocate* and *Hotel Reviews*, we preprocess them in the same way as Huang et al. (2021). The maximum text length is set to 256. More statistics of the datasets are in Table 8.

Some previous methods needs very careful hyper-parameter tuning. To make fair comparisons, most results of the baselines are copied from previous papers. But some settings are not unified, so we also reimplement them according to their source codes.

For DMR, we adopt its source code and adjust its sparsity constraint to get a sparsity similar to the annotated rationales. For A2R, we re-implement it to do token-level selection as other methods do.

The hyper-parameters of reimplemented models are manually tuned multiple times to get the best results. For our MGR, the early stopping technique is conducted according to the predictive accuracy of the development set. For our reimplemented DMR and A2R, although we have tried our best to tune the hyper-parameters, chances are that the hyper-parameters are not the best. To compensate for this potential issue, we do the test after every training epoch and choose their best results when they get the best $F1$ score on the test set.

The random seed is kept the same across all the experiments rather than manually selected. We think the experiments with one same random seed on multiple different settings and different datasets are enough to show the stability of our method. We also provide the standard deviations with running the experiments in Table 3 with five different ran-
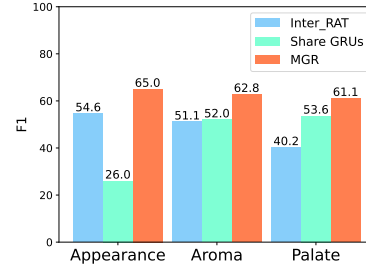

Figure 5: The comparison of sharing the encoders (GRUs) of MGR's generator. The rationale sparsity is around 10%.
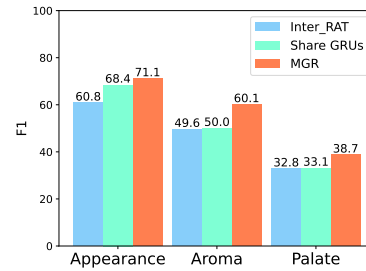

Figure 6: The comparison of sharing the encoders (GRUs) of MGR's generator. The rationale sparsity is around 30%.

dom seeds. The standard deviations are shown in Table 9.

A.2   Discussion on BERT encoder

In the field of rationalization, researchers generally focus on frameworks of the models and the methodology. Methods most related to our work do not use Bert or other pre-trained encoders (Chang et al., 2019, 2020; Huang et al., 2021; Yu et al., 2019, 2021; Yue et al., 2023). We use GRUs and GloVe to ensure the same experimental setup as our baselines for a fair comparison.

More importantly, how to finetune large models on the rationalization framework is still a significant challenge. Some recent studies (Chen et al., 2022) show that the methods with BERT encoders perform much worse than those with simple GRUs on BeerAdvocate and HotelReviews, which is shown in Table 10. VIB and SPECTRA are two RNP-based model. When using BERT, these two methods perform much worse than the vanilla RNP with GRUs (as compared to the results in Table 4(a)).

Methods	Appearance					Aroma					Palate				
	S	Acc	P	R	F1	S	Acc	P	R	F1	S	Acc	P	R	F1
MGR(Table 3)	10.9	80.5	87.5	51.7	65.0	10.3	89.7	78.7	52.2	62.8	10.8	86.0	65.6	57.1	61.1
MGR$_{\pm std}$	11.0$_{\pm 0.1}$	80.1$_{\pm 0.7}$	85.6$_{\pm 1.4}$	50.9$_{\pm 0.9}$	63.8$_{\pm 1.0}$	9.7$_{\pm 0.5}$	88.2$_{\pm 1.7}$	80.6$_{\pm 2.7}$	50.3$_{\pm 1.6}$	61.9$_{\pm 1.1}$	10.6$_{\pm 0.2}$	84.9$_{\pm 1.0}$	62.8$_{\pm 2.2}$	53.5$_{\pm 2.4}$	57.9$_{\pm 2.3}$
MGR(Table 3)	20.3	85.6	76.3	83.6	79.8	19.7	89.6	64.4	81.3	71.9	19.3	89.3	47.1	73.1	57.3
MGR$_{\pm std}$	19.8$_{\pm 0.3}$	86.7$_{\pm 1.1}$	79.4$_{\pm 1.9}$	84.9$_{\pm 1.2}$	82.1$_{\pm 1.5}$	19.3$_{\pm 0.3}$	88.6$_{\pm 0.8}$	65.8$_{\pm 0.9}$	81.3$_{\pm 0.8}$	72.7$_{\pm 0.6}$	19.6$_{\pm 0.7}$	88.4$_{\pm 1.1}$	46.3$_{\pm 1.9}$	72.8$_{\pm 1.6}$	56.6$_{\pm 1.8}$
MGR(Table 3)	30.4	88.5	57.2	93.9	71.1	29.8	91.6	45.8	87.4	60.1	30.3	89.3	27.3	66.5	38.7
MGR$_{\pm std}$	29.4$_{\pm 0.6}$	87.0$_{\pm 1.5}$	57.8$_{\pm 0.4}$	91.5$_{\pm 1.4}$	70.8$_{\pm 0.3}$	29.6$_{\pm 0.4}$	89.5$_{\pm 1.5}$	46.5$_{\pm 1.0}$	88.8$_{\pm 1.8}$	61.0$_{\pm 1.1}$	29.9$_{\pm 0.9}$	88.3$_{\pm 1.6}$	26.4$_{\pm 1.1}$	63.5$_{\pm 2.2}$	37.3$_{\pm 1.4}$

Table 9: The standard deviations of MGR on *correlated BeerAdvocate* with five different random seeds.

Methods	Beer-Appearance	Hotel-Cleanliness
VIB	20.5	23.5
SPECTRA	28.6	19.5

Table 10: Results with BERT. VIB: Paranjape et al. (2020), SPECTRA: Guerreiro and Martins (2021). The results are from Table 4 of Chen et al. (2022). The metric is F1 score.

A.3 The details of Beer-Skewed

The experiment was first designed by Yu et al. (2021). It deliberately induces degeneration to show the robustness of A2R compared to RNP. The predictor is first pre-trained using the first sentence of each text for a few epochs. In Beer Reviews, the first sentence is usually about appearance. So, the predictor will overfit to the aspect of *Appearance*, which is uninformative for *Aroma* and *Palate*. In fact, as compared to degeneration, we think it's more like spurious correlation, which may explain why A2R also fails in this situation.

A.4 More Results about Sharing the GRUs

Figure 4 in the main paper has shown the results of sharing the generators' encoders with the rationale sparsity being around 20%, and we further show the results with the sparsity being around 10% and 30% in Figure 5 and 6, respectively. Simply sharing the encoders may not be the best way to reduce the computational costs due to the damage on the model performance, but it still outperform Inter_RAT in most cases. For *Appearance* with 10% sparsity, the reason for the poor performance may come from two aspects. First, as compared to the percentage of human-annotated rationales (18.4%), 10% is too small. It is hard to find the true rationales under such sparsity constraint. Second, the shared encoder limits the explore power of MGR, making the above problem more severe. We will look for better method to reduce the computational costs in the future.

A.5 The rationale-overlap between different generators

Corresponding to Figure 3(b), we plot the rationale-overlap between different generators in Figure 7. The metric is $\frac{\|M_i - M_j\|_1}{\|M_i\|_1 + \|M_j\|_1}$, which represents the percentage of different tokens in rationales from different generators. M_i represents the binary rationale mask from the i-th generator. The figures show that the variance is high initially and gradually converges to a small value. So, the generators are diverse initially and finally converge to be the same.

B Proofs of Theorems

B.1 Derivation of Equation 6

To make the presentation succinct, we first discuss the case where n is an odd number.

$$\begin{aligned}
&\lim_{n \to \infty} p(k < n - k) \\
=&\lim_{n \to \infty} p(k < \frac{n}{2}) \\
=&\lim_{n \to \infty} \sum_{k=0}^{(n-1)/2} (C_n^k \cdot P_c^k \cdot (1 - P_c)^{n-k})
\end{aligned} \quad (16)$$

Since $P_c > 0.5$, we then have

$$\lim_{n \to \infty} \sum_{k=0}^{(n-1)/2} (C_n^k \cdot P_c^k \cdot (1 - P_c)^{n-k}) = 0. \quad (17)$$

There is nothing different expect that the upper limit of the summation should be replaced by $n/2 - 1$ when n is an even number.

B.2 Proof of Lemma 1

According to Equation 5, we have

$$\begin{aligned}
\frac{\partial R_P(\alpha)}{\partial \alpha} &= a \cdot k - (n-k)a - k \cdot b + (n-k)b \\
&= a \cdot (k - (n-k)) - b \cdot (k - (n-k)) \\
&= (a - b)(k - (n-k)).
\end{aligned} \quad (18)$$

Since we have $a > b$ and $k > n - k$, we get $\frac{\partial R_P(\alpha)}{\partial \alpha} > 0$. It means that to get a higher payoff, the predictor needs to increase α, i.e., it needs to move towards θ_P^1. The proof of Lemma 1 is completed.

Full	Decorrelated	Correlated	P_c
30564	15169	15395	0.67

Table 11: The P_c approximated by the statistical data of the *Beer-Aroma* dataset. It is approximated by $1 - \frac{Correlated}{Correlated*2+Decorrelated}$. We only count samples with negative labels because because the original dataset is unbalanced and we do sampling balance according to the number of negative samples during training.

B.3 Proof of Theorem 1

The proof is obvious. It's equal to that $\lim_{n \to \infty} p_{MGR}(spu) = 0$. The left derivation is the same as Appendix B.1.

B.4 Discussion about $P_c > 0.5$

For a dataset, there are some samples that contain both the causality and the spurious correlation (i.e., X_1 and X_2, corresponding to the number of *Correlated* in Table 11), and the other samples contain only the causality (i.e., X_1, corresponding to the number of *Decorrelated* in Table 11). So we always have the number of X_1 is larger than that of X_2. And for random selection, the probability of selecting X_1 is higher than selecting X_2, which means that $P_c > 0.5$. In Table 11, we approximate P_c by

$$
\begin{aligned}
P_c &= \frac{\text{Number}(X_1)}{\text{Number}(X_1) + \text{Number}(X_2)} \\
&= \frac{Decorrelated + Correlated}{Decorrelated + 2 * Correlated} > 0.5 \\
&= 1 - \frac{Correlated}{Correlated * 2 + Decorrelated} > 0.5.
\end{aligned}
\tag{19}
$$

B.5 Proof of Theorem 2

We first proof the left inequality of Theorem 2. For any two random variable Z_i, Z_j, we have

$$
\begin{aligned}
H(Z_i|Z_j) &= H(Z_i, Z_j) - H(Z_j) \\
H(Z_i|Z_j) &\geq 0 \\
H(Z_j) &\geq 0.
\end{aligned}
\tag{20}
$$

So, we have

$$
H(Z_i, Z_j) \geq H(Z_j), \tag{21}
$$

where the equality holds if and only if $H(Z_i|Z_j) = 0$, i.e., $Z_i = Z_j$. There is nothing different for $H(Z_i, Z_j) \geq H(Z_i)$. Then we easily get the left inequality of Theorem 2 through Mathematical Induction.

Then we proof the right inequality of Theorem 2. We first have

$$
\begin{aligned}
I(Z_i, Z_j) &= H(Z_i) + H(Z_j) - H(Z_i, Z_j) \\
I(Z_i, Z_j) &\geq 0,
\end{aligned}
\tag{22}
$$

where $I(Z_i, Z_j)$ is the mutual information. $I(Z_i, Z_j) = 0$ if and only if $Z_i \perp\!\!\!\perp Z_j$. So, we have

$$
H(Z_i, Z_j) \leq H(Z_i) + H(Z_j), \tag{23}
$$

with the equality holds if and only if $Z_i \perp\!\!\!\perp Z_j$. Then we easily get the right inequality of Theorem 2 through Mathematical Induction.
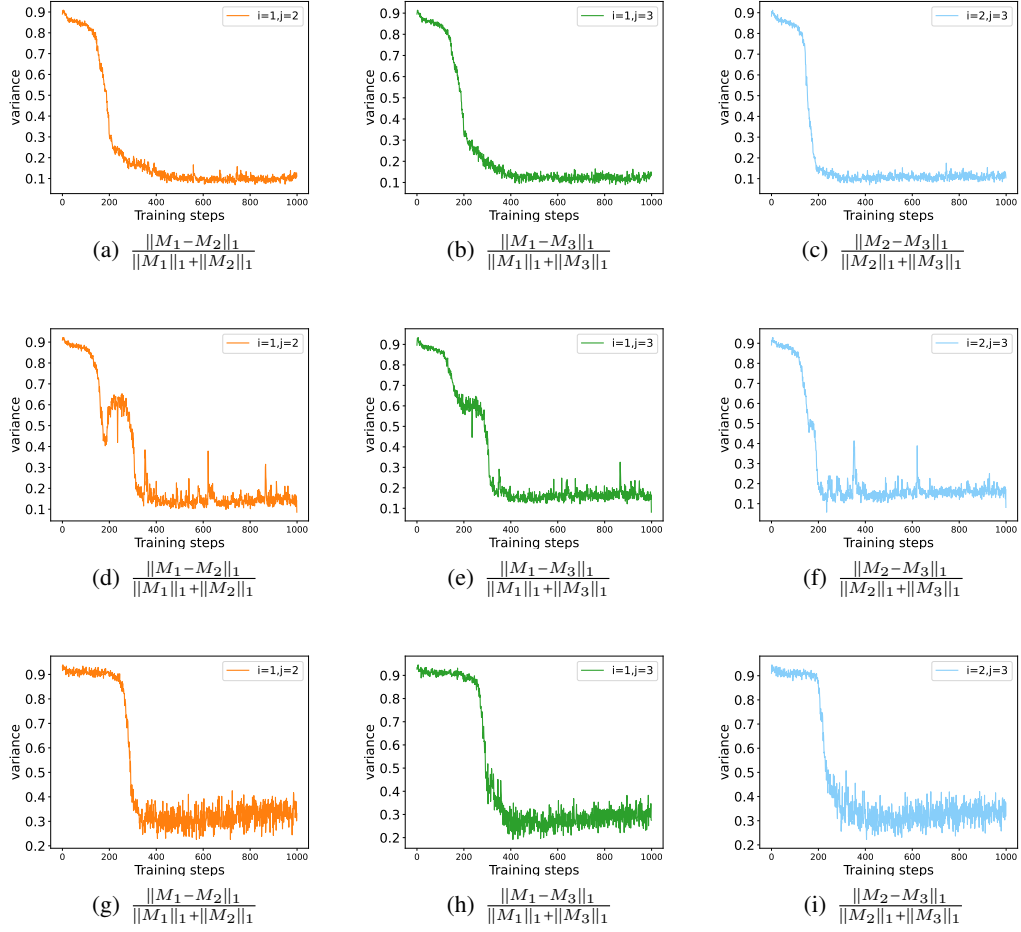
The proof of Theorem 2 is completed.

Figure 7: Rationale overlap from multiple generators on *decorrelated BeerAdvocate*. (a)(b)(c): Appearance aspect. (d)(e)(f): Aroma aspect. (h)(i)(j):Palate aspect. M_i, M_j: rationale masks (Equation 1) from the i-th and $j - th$ generators. **The y-axis shows the percentage of different tokens** in rationales from different generators.