The Impact of Integration Step on Integrated Gradients

Masahiro Makino[1] **Yuya Asazuma**[1,2] **Shota Sasaki**[3,1] **Jun Suzuki**[1,2]
[1]Tohoku University [2]RIKEN [3]CyberAgent, Inc.
{masahiro.makino.r6, asazuma.yuya.r7}@dc.tohoku.ac.jp
sasaki_shota@cyberagent.co.jp jun.suzuki@tohoku.ac.jp

Abstract

Integrated Gradients (IG) serve as a potent tool for explaining the internal structure of a language model. The calculation of IG requires numerical integration, wherein the number of steps serves as a critical hyperparameter. The step count can drastically alter the results, inducing considerable errors in interpretability. To scrutinize the effect of step variation on IG, we measured the difference between theoretical and observed IG totals for each step amount. Our findings indicate that the ideal number of steps to maintain minimal error varies from instance to instance. Consequently, we advocate for customizing the step count for each instance. Our study is the first to quantitatively analyze the variation of IG values with the number of steps.

1 Introduction

Researchers have focused on Explainable AI (XAI), which aims to provide insights into model behavior and predictions. One popular XAI method is feature attribution (Islam et al., 2021), generally referring to techniques that clarify why each feature was influential in determining the model's prediction.

Integrated gradients (IG) (Sundararajan et al., 2017) is one of the well-known feature attribution approaches and has been widely used in image (Adebayo et al., 2020; Kapishnikov et al., 2019) and language processing (Sanyal and Ren, 2021b; Sikdar et al., 2021) due to the many desirable explanation axioms and ease of gradient computation (Sanyal and Ren, 2021b). In recent years, IG has been applied to analyze language models (Kobayashi et al., 2023), and efforts have been made to enhance its performance specifically for language processing tasks (Sanyal and Ren, 2021b; Sikdar et al., 2021; Enguehard, 2023).

In IG, a property known as *completeness* (Sundararajan et al., 2017) posits that the sum of the contributions of each feature equals the difference between the output and the sum. This fundamental property offers a way of interpreting the value of each contribution as its influence on the output as follows Eq. 2. It also quantifies each contribution value relative to the output, enabling comparisons between contributions.

However, *completeness* is often violated because the numerical integration required to compute IG introduces errors. *Completenes* violation compromises the interpretability of the contributions and the results obtained from comparisons among the contributions. We have also identified instances where errors adversely affect the interpretation of the contributions (see Figure 2). Hence, to ensure the reliability and accuracy of IG, it is vital to determine the number of steps to minimize such errors properly.

Given these factors, it is necessary to ensure an adequate number of steps to reduce errors to guarantee IG's reliability. However, as shown in Table 1, researchers often subjectively set the number of steps to use IG for each model or dataset. Several references address this issue (Sundararajan et al., 2017), requiring between 20 and 300 steps for a sentence classification task using a CNN model (Kim, 2014) and between 100 and 1000 steps for a translation task using LSTM (Wu et al., 2016). Nevertheless, there has yet to be a quantitative analysis that can be sufficient regarding the number of steps. In addition, no studies specifically address the number of steps required for modern language models (LMs) such as BERT (Devlin et al., 2019).

Therefore, in this study, we measured the error between the theoretical and measured values of the total IG sum at each number of steps to quantitatively analyze the change in the contribution value depending on the number of IG steps in the LM. The results show that the ideal number of steps that minimize the error varies from instance to instance,

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 279–289
March 21-22, 2024 ©2024 Association for Computational Linguistics

Table 1: **Number of steps set when using IG in text classification.** In previous research, the number of steps is set for each model and not for each instance.

Step	Model	Paper
50	CNN	(Liu and Avci, 2019) (Dixon et al., 2018)
50, 250	DistilBERT, RoBERTa, BERT	(Enguehard, 2023)
10, 30, 100, 300	DistilBERT, RoBERTa, BERT	(Sanyal and Ren, 2021a)
1000	Linear / Logistic regression	(Han et al., 2022)
100, 1000	BERT, LSTM	(Bastings et al., 2022)

even for the same dataset model. This result argues that the number of steps should be set on an instance-by-instance basis. Our study is the first to quantitatively analyze the variation of IG values with the number of steps.

2 Integrated Gradients

The method of generating post-hoc explanations for each model output is known as feature attribution (Simonyan et al., 2014). This method allows for the assessment of the contribution of input features to the prediction results of machine learning models. It provides insights into how much a model's predictions rely on specific features.

Integrated gradients (IG) (Sundararajan et al., 2017) is a type of feature attribution method. IG is popular over other feature attribution methods due to its simplicity, relatively low computational cost, and adherence to mathematically rigorous axioms (Lundstrom et al., 2022).

In the field of NLP, IG has proven valuable, with researchers developing enhanced methods tailored to language-specific tasks (Sanyal and Ren, 2021b; Sikdar et al., 2021; Enguehard, 2023) and utilizing it for LM analysis (Kobayashi et al., 2023).

The IG formula for an input $\boldsymbol{x}$ along the i-th dimension is as follows:

$$\mathrm{IG}_i(\boldsymbol{x}) = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial F}{\partial x_i}(\boldsymbol{x}' + \alpha(\boldsymbol{x} - \boldsymbol{x}'))d\alpha. \tag{1}$$

Here, F is the deep neural network, $\boldsymbol{x}'$ is a baseline embedding along the i-th dimension, and α is the variable of integration.

The IG calculation involves sampling along a linear path from the baseline vector to the input vector and computing and integrating the gradient for each sample. Here, the sampling points are determined by the numerical integration method and the number of steps. The number of steps is a vital hyperparameter that determines the integration accuracy. However, many steps require much backpropagation, resulting in high computational costs.

2.1 Completeness Axiom

The *completeness* axiom (Sundararajan et al., 2017) is one of the several mathematical principles IG satisfies, indicating that the sum of IG in each dimension is the model output value for the given input minus the model output value for the baseline as follows:

$$\sum_{i=1}^{n} \mathrm{IG}_i(\boldsymbol{x}) = F(\boldsymbol{x}) - F(\boldsymbol{x}'). \tag{2}$$

2.2 Issues in setting the number of steps

In practical applications, there are cases where this axiom does not hold due to errors caused by numerical integration. To prevent the effects of errors, a sufficiently large number of steps must be set in advance.

For instance, in the sentence classification task of a CNN model, Sundararajan et al. (2017) argues that the number of steps should range from 20 to 300, while for LSTM translation tasks, it should be within 100 to 1000. However, research has not identified what constitutes sufficient error reduction for practical application in Table 1. The ideal number of steps may vary depending on the model and dataset, and there needs to be a discussion of the number of steps in LM, like in BERT. Therefore, we analyze the impact of the number of steps on IG values in LM.

3 Experimental Settings

3.1 Verification Indicators

Approximation error (AE) measures the deviation of the actual measured value from the theoretical value, and we compute the error for each step as follows:

$$\mathrm{AE} = \left| \frac{\sum_i \widetilde{\mathrm{IG}}_i(\boldsymbol{x}) - (F(\boldsymbol{x}) - F(\boldsymbol{x}'))}{F(\boldsymbol{x}) - F(\boldsymbol{x}')} \right| \tag{3}$$

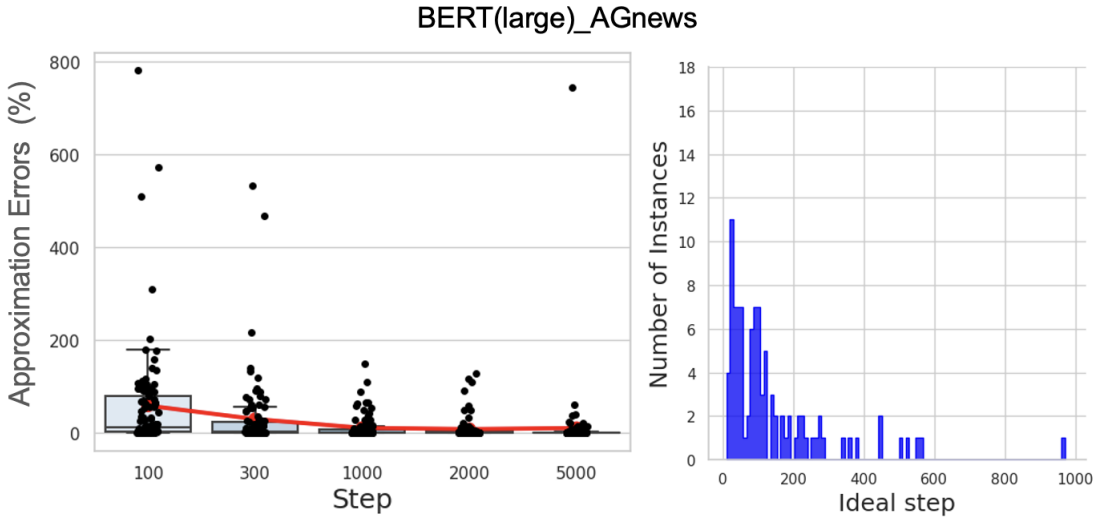

Figure 1: **Boxplot on the left**: The red line represents the approximation errors average for each number of steps, and a single point represents the approximation errors for a single instance. **Histogram on the right**: The number of steps ideal for each instance. The vertical axis is the number of instances with the ideal number of steps on the horizontal axis. It can be seen that the ideal number of steps is different for each number of instances. However, nearly 60% of the instances had an ideal number of steps within 100 steps.

Here, $\sum_i \widetilde{\mathrm{IG}}_i$ is the sum of the measured IG calculated by numerical integration.

The AE reflects the discrepancy between the theoretical sum value of IG and the actual measured value. Also, Sundararajan et al. (2017) argues that the number of steps should be adjusted based on the AE.

3.2 Baseline Vector

IG's baseline vector remains an ongoing discussion in the field (Sturmfels et al., 2020; Tan, 2023; Bastings et al., 2022). In our experiments, we align with the notion that the baseline vector should possess minimal information for the model and use the maximum entropy baseline as the baseline vector (Tan, 2023). This vector exhibits the most uniformly distributed model outputs in the test dataset.

3.3 Dataset & Model

We use AG News (Gulli., 2004), 20 News (Ko, 2012), and SST-2 (Socher et al., 2013) as our datasets. These datasets are widely used in sentence classification. Details of the datasets are available in Appendix A.1. BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) serve as the LMs for the experiment, utilizing both base and large models. Details of the models are available in Appendix A.2.

3.4 Other experimental settings

We used Riemann sum and Gauss-Legendre integration as our numerical integration methods.

In the interest of realistic experimental timeframes, we randomly sampled 100 instances from the test data for each dataset.

4 Experimental Results

Since the Riemann sum results were consistently better than the Gauss-Legendre integration results, we report the Riemann sum results in the following experiments. See Appendix A.4 for details. Here, we show the case of the BERT(large)-AGnews model, but results for other models are given in Appendix A.7.

4.1 Quantitative Analysis of Errors

We performed a quantitative analysis to investigate the potential errors that can arise if IG is calculated for all instances at a specified fixed number of steps. We calculated the IG values for all instances at step numbers 100, 300, 1000, 2000, and 5000 steps. We then calculated the approximation errors (AE) for each instance at each step number to review how the AE would perform if the same number of steps were applied across each instance.

Error for each step From the results of the box plot on the left in Figure 1, we observed that even with a vast number of steps (>1000), the AE are

Step	Error	Visualization
1000step	Error = **25%**	Ġpeople Ġhave Ġbeen Ġkilled Ġin ĠKashmir Ġin Ġan Ġincrease
140step	Error = **0.62%**	Ġpeople Ġhave Ġbeen Ġkilled Ġin ĠKashmir Ġin Ġan Ġincrease
1000step	Error = **17%**	make sure the bike has cooled at least 6 hours since being run
130step	Error = **0.94%**	make sure the bike has cooled at least 6 hours since being run
1000step	Error = **59%**	russian oil giant si ##bn ##eft today rejected
520step	Error = **4.8%**	russian oil giant si ##bn ##eft today rejected

Figure 2: **Visualization of IG.** Above each line is the visualization using the assumed fixed-step. Below each line is a visualization of when the ideal step is used. From top to bottom, visualization are RoBERTa(base)_AGnews, BERT(large)_20news and BERT(large)_AGnews.

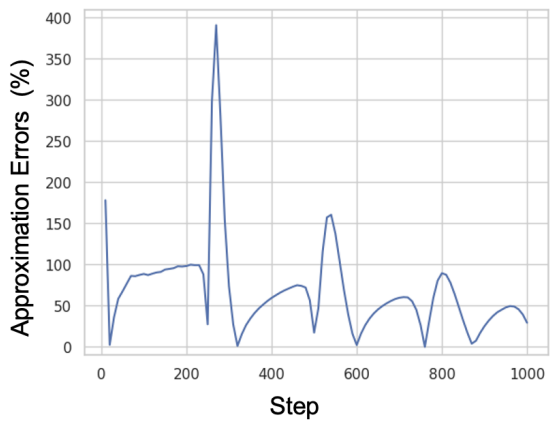

Figure 3: **Approximation errors per step for an instance.** Increasing the number of steps does not lead to a monotonic decrease in the approximation errors.

instances with significant errors. There are several samples with AE greater than 100%, and even using 5000 steps, there are instances with AE more significant than 700%. These results indicate that larger steps cannot guarantee fewer errors. The ideal number of steps may vary from instance to instance.

Also, in Figure 3, we observed instances where the AE moved up and down sharply as the steps increased. This case shows that the AE increases with an increase in the number of steps. It was confirmed that the AE does not decrease monotonically with the increase in the number of steps.

Ideal step Therefore, we investigated the distribution of the ideal number of steps for each instance. The ideal number of steps here is defined as the number of steps for which the AE is initially within 5%. This definition is informed by the number proposed (Sundararajan et al., 2017).

From the right histogram in Figure 1, we observed 98 out of 100 instances with an ideal number of steps within the 1000 steps. Of these, more than 60 instances had the ideal number of steps within 100 steps. This result indicates that even for LMs such as BERT, even a small number of steps, as small as 100, is sufficient for convergence in more than half of the instances. In contrast, even 1000 steps cannot guarantee convergence for all instances.

This result suggests that fixing the number of steps for all instances may not be ideal for error reduction.

4.2 Qualitative Analysis of Errors

Since the experiments in Section 4.1 revealed that some instances do not converge in error even with significant steps, we perform a qualitative analysis for those instances where the error does not converge. The visualization rules are those outlined in (Sundararajan et al., 2017). See Appendix A.6 for details.

Visualization From the visualization results in Figure 2, it can be confirmed that in instances where errors occur, the contribution values do not change in all samples, but rather the values change significantly, concentrating on certain features. In addition, errors are caused by the observation of non-existent contribution values. From this, it can be inferred that a significant error is caused by erroneous numerical integration for contributions that have an oscillating shape, although the sum is zero for the entire interval.

5 Why is the ideal number of steps different for each instance?

This section explores why is the ideal number of steps different for each instance. As the basis for this discussion, we focus on α in the Eq. 1. For each minuscule change in α, the gradient is calculated, and ultimately, the gradient is integrated. The larger the number of steps, the more minute the changes in α, enabling a more detailed computation and integration of the gradient.

Each instance has a gradient of zero for most segments, and only at certain points does the gradient change significantly. The point at which this gradient changes significantly varies greatly from instance to instance. This point of pronounced gradient change fluctuates at a specific α value, a phenomenon common in the imaging field. If these crucial points of gradient change are not accurately captured, it becomes impossible to calculate integral parts of the IG sum.

To illustrate, consider a 10-step integration where the gradient is computed for each $alpha$ of values in 0.1 increments from 0 1.0. If there are significant changes in the gradient at any of these α values, the IG error will be small. But, if there are no substantial changes in the gradient for any of these α values and a significant shift happens, say, at 0.15, then the IG error will be considerable because the exact gradient value at this point cannot be calculated. From this, it can be inferred that instances requiring a smaller, ideal number of steps have a narrower range of α values where a substantial change in the gradient occurs.

Figure 4 to the left illustrates the gradient per α for an instance where the error is maximized at 270 steps and minimized at 870 steps. Clearly, the substantial gradient captured at 870 steps is missed at 270 steps. Figure 4 to the right, on the other hand, presents the gradient per α for instances where the error is relatively small for both 270 steps and 870 steps. In this case, it is evident that the gradient is adequately captured at both 870 steps and 270 steps.

As these instances suggest, the ideal number of steps varies per instance because the locations of large gradients and the size of these locations differ across instances.

6 Discussion

Our analysis reveals that the number of integration steps required for each instance is different.

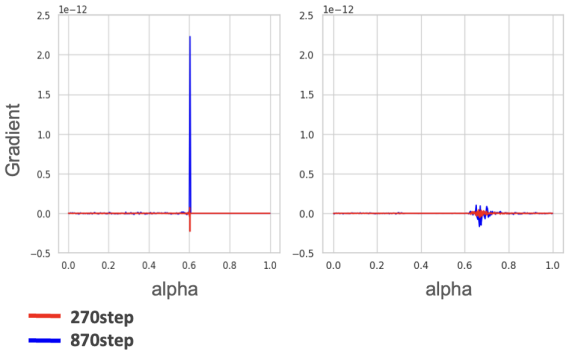

Figure 4: **Gradient value per α.** The red line is the gradient value for each α when done in 270 steps. The blue line is the gradient value for each α when done in 870 steps.

Therefore, we recommend that the number of steps be increased sequentially until it falls below a specific error, thereby reducing the error. For example, we can ensure that the IG satisfies *completeness* by initially setting the number of steps to 2^n and running with increasing n until the error converges to a constant.

Optimizing the number of steps on an instance-by-instance basis would also make IG more efficient since our analysis has shown that the number of steps required is negligible for many instances (Figure 1). However, we keep this part as a future study since constructing a methodology to find better solutions.

7 Conclusion

The researcher subjectively determines the number of steps in IG for each dataset and model, which raises questions about the reliability of IG.

In this study, we quantitatively analyzed the error for each number of steps. As a result, half of the instances in which the appropriate number of steps is around 100 steps, but on the other hand, instances in which the error does not converge even at 1000 steps or more were confirmed.

These results indicate that the current mainstream method of fixing the number of steps for each model or data set runs the risk of producing instances with broken contributions and undermining the reliability of IG's analysis results. To solve this, we also proposed to change the integration step for each instance.

Our study is the first to quantitatively analyze the variation of IG values with the number of steps and to identify problems with existing integration methods.

Limitations

In this experiment, 100 instances were randomly selected for each combination of model and dataset. This selection was necessary due to the computational cost factor. Further investigation involving more instances is needed for more accurate experiments.

The maximum entropy vector was used as the baseline for this experiment. Future validation using different baseline vectors is needed for a comprehensive model performance evaluation under various baselines.

In our validation, we used the correct Riemann sum. Future analysis using multiple Riemann sums, such as left Riemann sums and midpoint Riemann sums, is needed.

When ensuring the number of steps for each instance, the cumulative number of steps and the computational cost may increase, which is a potential issue.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers JP21H04901 and JST Moonshot R&D Grant Number JPMJMS2011-35 (fundamental research).

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2020. Sanity checks for saliency maps.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. "will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, page 67–73, New York, NY, USA. Association for Computing Machinery.

Joseph Enguehard. 2023. Sequential integrated gradients: a simple but effective method for explaining language models.

FacebookInc. 2023. Captum. https://captum.ai/.

Antonio Gulli. 2004. Agnews. http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html.

Tessa Han, Suraj Srinivas, and Himabindu Lakkaraju. 2022. Which explanation should i choose? a function approximation perspective to characterizing post hoc explanations.

Sheikh Rabiul Islam, William Eberle, Sheikh Khaled Ghafoor, and Mohiuddin Ahmed. 2021. Explainable artificial intelligence approaches: A survey. *ArXiv*, abs/2101.09429.

Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. 2019. Xrai: Better attributions through regions.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Youngjoong Ko. 2012. A study of term weighting schemes using class information for text classification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 1029–1030, New York, NY, USA. Association for Computing Machinery.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Feed-forward blocks control contextualization in masked language models.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Daniel D Lundstrom, Tianjian Huang, and Meisam Razaviyayn. 2022. A rigorous study of integrated gradients method and extensions to internal neuron attributions. 162:14485–14508.

Soumya Sanyal and Xiang Ren. 2021a. Discretized integrated gradients for explaining language models.

Soumya Sanyal and Xiang Ren. 2021b. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. 2021. Integrated directional gradients: Feature interaction attribution for neural NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 865–878, Online. Association for Computational Linguistics.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Pascal Sturmfels, Scott Lundberg, and Su-In Lee. 2020. Visualizing the impact of feature attribution baselines. *Distill*. Https://distill.pub/2020/attribution-baselines.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks.

Hanxiao Tan. 2023. Maximum entropy baseline for integrated gradients. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

A Appendix

A.1 Datasets

The datasets we used are widely used in NLP classification problems.

AG News News articles are grouped into four main categories ("Sports," "Business," "Science/Technology," and "Entertainment") (Gulli., 2004).

20 News News articles are grouped into 20 categories ("Computers," "Science," "Sports," "Politics," and more) (Ko, 2012).

SST-2 The Stanford Sentiment Treebank-2 is provided for sentences with positive or negative emotional polarity (Socher et al., 2013).

Table 2: Datil of datasets

Dataset	**train / test**	**class label**	**max lengths**
AG news	120k / 7.6k	4 classes	50
20 news	11.3k / 7.53k	20 classes	200
SST2	6.92k / 1.82k	2 classes	20

A.2 Models

BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) serve as the LMs for the experiment, utilizing both base and large models. A linear layer is affixed to these models as the final layer, and a softmax function is employed to address the sentence classification task.

BERT BERT uses a bidirectional transformer architecture. Unlike regular language models, it considers both left and right contexts simultaneously to understand words in context (Devlin et al., 2019).

RoBERTa RoBERTa is a version that takes the architecture and basic ideas of BERT and optimizes the way the model is trained (Liu et al., 2019).

Table 3: Accuracy of test data

Model	**Accuracy**		
	AG News	**20 News**	**SST-2**
BERT-base(110M)	0.94	0.64	0.86
BERT-large(340M)	0.93	0.65	0.87
RoBERTa-base(125M)	0.94	0.61	0.88
RoBERTa-large(561M)	0.93	0.64	0.88

A.3 Integration Method

Numerical integrals are pivotal for IG. The library Captum (FacebookInc., 2023), a comprehensive Pytorch implementation of XAI methods, employs Riemann Sum and Gauss-Legendre integrals for IG's numerical integration.

Riemann Sum The Riemann Sum is a technique used to approximate the area under a function.

When applying the Riemann sum to IG or an input x along the i_{th} dimension, the approximation can be expressed as follows:

$$\mathrm{IG}_i^{\mathrm{approx}}(x) = (x_i - x_i') \sum_{k=0}^{n} \frac{\partial F}{\partial x_i}(x' + \frac{k}{n}(x - x'))\frac{1}{n}, \quad (4)$$

where F represents the deep neural network, x' is a baseline embedding, and n is the sampling size. This equation allows for the estimation of the contribution of the i-th feature to the prediction results of the model.

Gauss-Legendre Integral Gauss-Legendre integral is a method used to approximate definite integrals, typically on the interval [-1, 1]. It involves finding the roots, denoted as x_k, of the n_{th} order Legendre polynomial, $P_n(x)$. These roots are the distinct real solutions of the polynomial of degree n that lie within the interval [-1, 1].

Applying the Gauss-Legendre integral to IG yields the following equation:

$$\mathrm{IG}_i^{\mathrm{approx}}(x) = \frac{(x_i - x_i')}{2} \sum_{k=1}^{n} \frac{\partial F}{\partial x_i}(x' + w_k(\frac{x_k}{2} + \frac{1}{2} - x')), \quad (5)$$

where, the weights, denoted as w_k, corresponding to each root x_k are computed.

This method allows us to approximate the integral of a function using a Legendre polynomial of the appropriate degree. Because the roots and weights of the Legendre polynomial satisfy certain conditions, this method is numerically very stable and can have high accuracy for integrals of high dimension and integrals of special functions.

A.4 Riemann sum vs Gauss-Legendre

Since the results for Riemann sum were consistently better than those for Gauss-Legendre integration, the results for Riemann sum are reported in Figure 5.

A.5 Ideal step of instances

The ideal step was defined as the number of steps that the error becomes within 5% for the first time

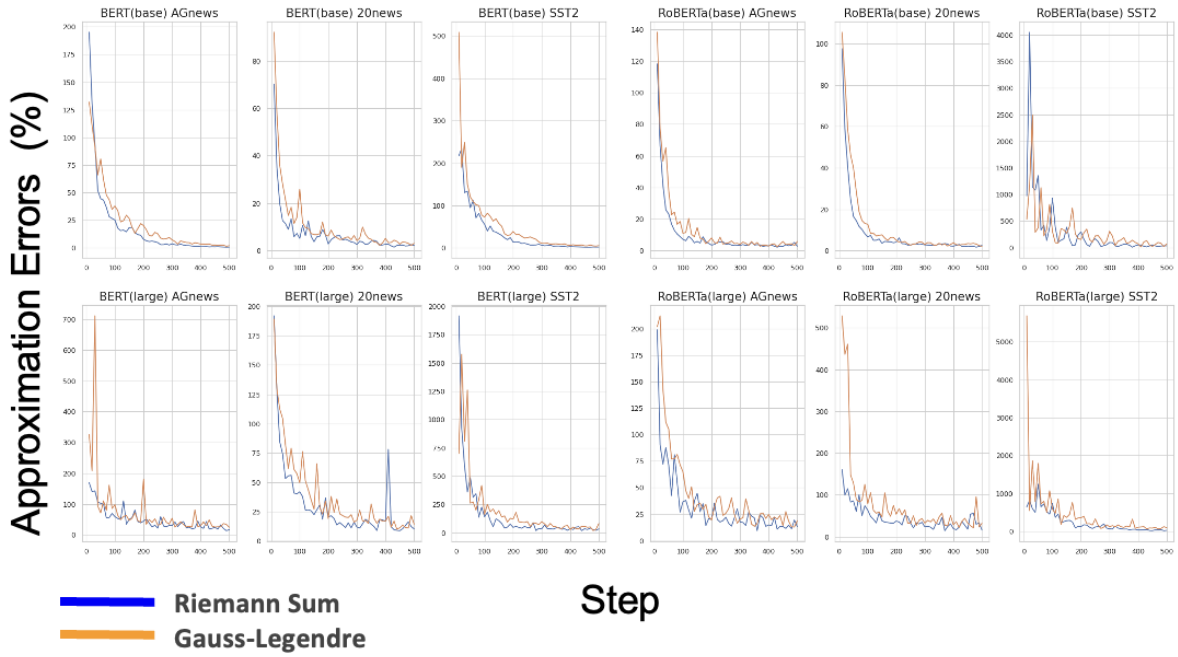

Figure 5: **Mean of Approximation errors by Riemann sum and Gauss-Legendre.**

by increasing the number of steps. Almost all instances had an ideal step within 1000 steps, but there were a few instances where the error was never within 5% within 1000 steps. For each model and data set, we described the number of instances in which the ideal step was within 1000 steps out of 100 instances being analyzed in 4.

Table 4: Number of instances having ideal steps within 1000 steps

Model	AG News	20 News	SST-2
BERT-base	100	100	100
BERT-large	98	99	99
RoBERTa-base	99	100	95
RoBERTa-large	99	100	97

A.6 Visualization rule

The appendices below detail the calculation of the contribution per word, which is obtained by summing the contributions calculated for each dimension corresponding to each word. The visualization rules align with those used in IG paper (Sundararajan et al., 2017). In these visualizations, green represents a positive contribution and red represents a negative contribution. The darkest shade is assigned to the most considerable absolute value of the contribution calculated for each word, and colors lighten as they approach zero.

A.7 Quantitative Analysis of Errors

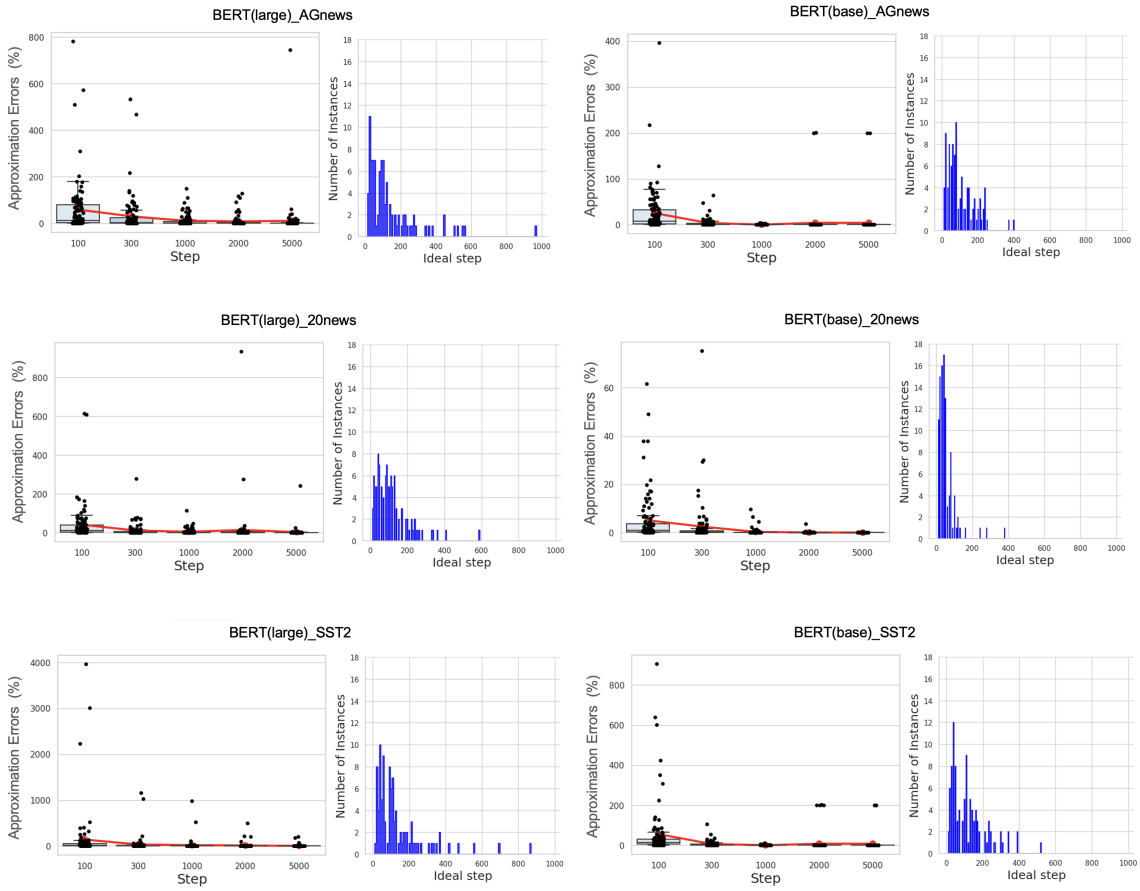

Figure 6: **BERT.** Boxplot on the left: The red line represents the approximation errors average for each number of steps, and a single point represents the approximation errors for a single instance. Histogram on the right: The number of steps ideal for each instance. The vertical axis is the number of instances with the ideal number of steps on the horizontal axis.

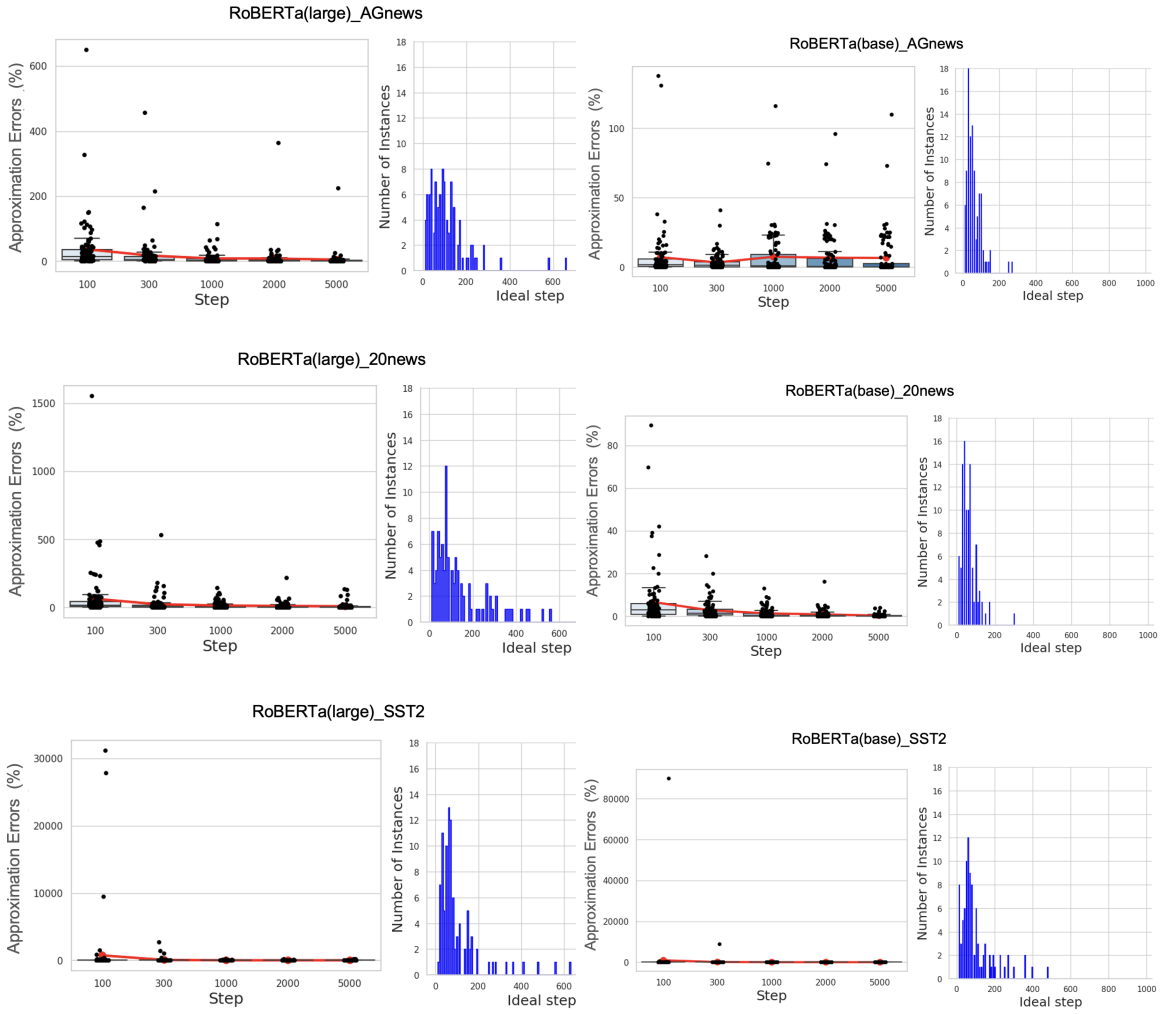

Figure 7: **RoBERTa.** Boxplot on the left: The red line represents the approximation errors average for each number of steps, and a single point represents the approximation errors for a single instance. Histogram on the right: The number of steps ideal for each instance. The vertical axis is the number of instances with the ideal number of steps on the horizontal axis.