lightX: Explanation-Guided Learning for Lightweight Models

Korea University COSE461 Final Project

Joonhyuk Kang
Department of Computer Science
Team 12
2021320141

Gyuwon Kim
Department of Computer Science
Team 12
2019140415

Dohyeok Kwon
Department of Computer Science
Team 12
2020320026

Yonggeun Kim
Department of Computer Science
Team 12
2022320097

Abstract

While large language models like BERT offer high accuracy, their computational demands limit their use in resource-constrained settings. In this work, we propose an explanation-guided learning framework that trains a lightweight MLP model using not only ground-truth labels, but also explanation patterns derived from a BERT teacher model via Integrated Gradients (IG). By combining cross-entropy loss with an explanation imitation loss, the student model learns to align both its predictions and its feature attributions with those of the teacher. Experiments on the AG News dataset show that this approach significantly improves MLP performance compared to a baseline trained only with labels. Notably, even when trained without labels, the model achieves meaningful accuracy using only explanation-based supervision. These results suggest that explanation signals can serve as an effective, lightweight form of knowledge transfer, improving model performance and interpretability without relying on complex architectures.

1 Introduction

Recent advances in large-scale pre-trained language models such as BERT [1] have led to remarkable improvements across a wide range of natural language processing (NLP) tasks. Despite their impressive accuracy, these models are computationally expensive and memory-intensive, making them impractical for deployment in resource-constrained environments.

To address this limitation, lightweight models such as multilayer perceptrons (MLPs) are often employed. However, these simpler architectures typically suffer from a substantial drop in performance when compared to their larger counterparts. Prior efforts to close this gap have focused on knowledge distillation techniques, in which the smaller model learns to mimic the output logits of the teacher model [2]. While effective, these approaches rely on the soft labels from the teacher and do not exploit the internal reasoning or explanation patterns that reflect how the teacher arrives at its predictions.

In contrast, our work explores a novel approach: rather than mimicking the teacher's predictions, we encourage the student model to align with the teacher's explanation. Specifically, we propose an *explanation loss* that guides an MLP classifier to imitate the token-level attribution patterns of a BERT model, as measured by Integrated Gradients (IG). These attributions reflect how much each

input token contributes to a specific class prediction, providing a rich supervisory signal beyond just the labels.

Our method combines the conventional cross-entropy loss computed from ground-truth labels with an explanation imitation loss that measures the discrepancy between the student's and teacher's IG attribution vectors.

Experiments on the AG News dataset demonstrate the effectiveness of our approach. With an optimal balance between the two losses (e.g., $\alpha = 0.1$), our model significantly outperforms a plain MLP baseline, achieving up to 82.71% accuracy while maintaining a lightweight footprint. Furthermore, even when trained with only the explanation loss ($\alpha = 0$), the model achieves competitive performance, suggesting that explanations alone can provide meaningful supervision.

These findings highlight the potential of explanation-based training as an alternative or complementary form of supervision for compact and interpretable models [3].

2 Related Work

2.1 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) refers to techniques that aim to make the decision-making process of AI models transparent by providing explanation for their predictions. As deep neural networks have grown increasingly complex, concerns have arisen around their opaque nature, particularly in sensitive domains such as healthcare, finance, and law. To address this, post-hoc explanation methods, including SHAP [4], LIME [5], and Integrated Gradients (IG) [6] have been proposed. These techniques provide quantitative estimates of how much each input feature contributes to the model's final prediction.

2.2 Integrated Gradients (IG)

Integrated Gradients (IG) [6] is a gradient-based attribution method that assigns importance scores to input features based on their contribution to the model's prediction. It computes the integral of the gradients along a linear path from a baseline input x' to the actual input x:

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

IG satisfies two desirable axioms: *sensitivity*, which ensures that important features receive non-zero attribution, and *implementation invariance*, meaning that functionally equivalent models yield the same explanations. These properties, along with its ease of implementation for transformer-based models such as BERT, have made IG a popular and reliable XAI method for both interpretation and training-time guidance.

2.3 Explanation-Guided Learning without Teacher Models

While XAI techniques are typically used for interpretation and debugging, there have also been attempts to incorporate them into the training of models. Mazumder et al. [7] proposed a framework where LIME explanations are used to guide incremental learning. Their method selectively reintroduces misclassified samples during training and adjusts their loss weights based on the Euclidean distance between the LIME vectors of the predicted and true classes. This strategy demonstrates that explanations can act as informative feedback for model refinement.

However, this method generates explanations from the same model that is being trained, limiting its capacity to provide reliable supervision. Without an external teacher model, explanation alignment risks reinforcing flawed reasoning, especially in complex tasks. Moreover, the approach is only applied to misclassified examples, reducing its influence on the model's overall learning dynamics.

2.4 Knowledge Distillation and DiXtill

Knowledge Distillation (KD) [8] is a widely used framework for compressing large models into smaller ones. It trains a student model to mimic the softmax outputs of a teacher model, allowing for better generalization than training on hard labels alone.

To go beyond output mimicry, Cantini et al. [9] proposed DiXtill, a framework that incorporates explanation alignment into the distillation process. In this method, Integrated Gradients (IG) are extracted from a BERT-based teacher model and aligned with attention scores from a Bi-LSTM student model using a cosine similarity loss.

However, this design has a key limitation due to the structural differences in how the teacher and student generate explanations. The teacher model uses IG, which calculates how important each input is by observing how changes in the input affect the model's prediction. In contrast, the student model produces its explanations using attention weights—trainable values that highlight which parts of the input are considered important during training.

Because IG and attention follow fundamentally different principles, using a simple cosine similarity to align them introduces a structural mismatch. This makes it difficult to ensure that the student's explanations are truly faithful or consistent with those of the teacher model.

3 Approach

The primary objective of this study is to train a lightweight Multi-Layer Perceptron (MLP) model that retains both the predictive performance and interpretability of a large-scale, computationally intensive pre-trained language model (LLM) such as BERT.

In contrast to Dixtill, which employs recurrent architectures and integrates explanation signals into hidden states, our method uses Integrated Gradients (IG) to compute feature attributions from both the teacher and the student models after their predictions are made. This design separates the explanation process from the model's internal architecture, allowing explanation-guided learning to be applied flexibly across a wide range of models, including simple feedforward networks such as MLPs.

Building on this principle, we introduce an explanation-guided learning framework that aligns the explanation pattern of a compact student model with that of a powerful teacher. Specifically, the target model (MLP) is trained using a custom loss function that jointly optimizes for both accurate predictions, based on ground-truth labels, and alignment with the explanation patterns generated by a BERT-based teacher model. This encourages the student model to not only replicate the teacher's outputs but also learn to ground its decisions in similar input features.

Our approach offers greater architectural flexibility while ensuring consistency in the decision-making process between student and teacher models.

3.1 Model Architecture

In this study, we build MLP model as a classifier that predicts class probabilities from input text. We compute two distinct loss components during the training of the MLP model :

1. $\mathcal{L}_{\text{CE}}$: cross-entropy loss between the MLP's softmax output and the ground-truth label
2. $\mathcal{L}_{\text{XAI}}$: explanation similarity loss between the explanation vectors of the MLP and BERT models.

The MLP is trained to minimize the total loss, which is a weighted combination of these two components. This encourages the MLP not only to mimic the prediction outcomes of the BERT model but also to replicate the way BERT arrives at its predictions—that is, the underlying explanation pattern. As a result, the learning framework is designed to preserve both predictive performance and explanation consistency.

To make this comparison meaningful, the input structures of both models must be aligned. Therefore, we use the same BERT tokenizer for both models to segment the input sentences into subword units. This ensures that token positions remain aligned between the two models, allowing the

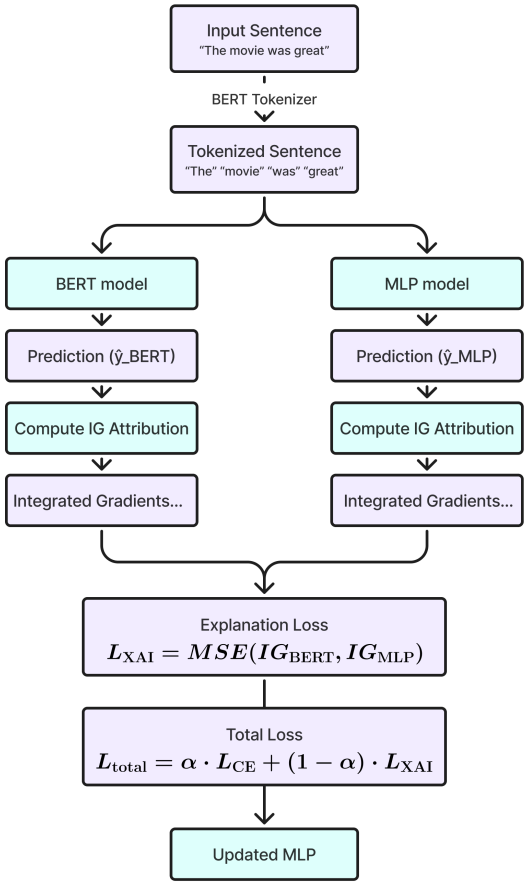

Figure 1: Diagram of the proposed explanation-guided training framework. The same input sentence is tokenized and fed into both a pretrained BERT model (teacher) and a lightweight MLP model (student). Each model produces a prediction and a corresponding attribution vector based on Integrated Gradients (IG), which are then used to compute the loss and update the MLP model accordingly.

explanation vectors computed using Integrated Gradients (IG) to be compared in a token-wise one-to-one correspondence.

3.1.1 Base Model (BERT)

The base model is based on the pre-trained `bert-base-uncased` model provided by HuggingFace. Input sentences are first tokenized into subword units using the BERT tokenizer, and the resulting token sequence, including the special `[CLS]` token, is passed into the BERT Transformer encoder. The final sentence representation is extracted from the output vector corresponding to the `[CLS]` position, denoted as $\mathbf{h}_{[\text{CLS}]}$, and passed through a linear classification layer followed by softmax to produce class probabilities:

$$\hat{\mathbf{y}}_{\text{BERT}} = \text{Softmax}(\mathbf{W}_{\text{BERT}} \cdot \mathbf{h}_{[\text{CLS}]} + \mathbf{b}_{\text{BERT}})$$

where $\mathbf{W}_{\text{BERT}}$ and $\mathbf{b}_{\text{BERT}}$ denote the weight matrix and bias of the classification layer, respectively.

As a result of applying Integrated Gradients (IG) to the predicted output, we obtain an explanation matrix $\mathbf{E}_{\text{BERT}} \in \mathbb{R}^{n \times C}$, where n is the number of input tokens and C is the number of target classes. This matrix reflects how much each input token contributes to the prediction of each class, and it is incorporated into the loss function to promote explanation consistency during training.

3.1.2 Target Model (MLP)

The target model is a lightweight multi-layer perceptron (MLP) that takes input sentences as token ID sequences. Each token ID is mapped to a dense vector representation via a learnable embedding matrix, which is randomly initialized and optimized during training.

The word embeddings for each token are averaged to obtain a single fixed-length sentence vector, which is then passed through two fully connected layers with a ReLU activation and dropout regularization:

$$\mathbf{h} = \text{ReLU}(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$

$$\hat{\mathbf{y}} = \text{Softmax}(\mathbf{W}_2 \cdot \text{Dropout}(\mathbf{h}) + \mathbf{b}_2)$$

This architecture significantly reduces the number of parameters and computational cost compared to the BERT model. In our study, the MLP is trained not only to replicate BERT's predictions, but also to approximate its explanatory behavior.

In addition to predicting class probabilities, the model also applies Integrated Gradients (IG) to the same input to generate an explanation matrix $\mathbf{E}_{\text{MLP}} \in \mathbb{R}^{n \times C}$, where n is the number of input tokens and C is the number of target classes. Each element in this matrix represents the contribution of a token to the prediction of each class. This explanation matrix is then compared with a reference explanation matrix $\mathbf{E}_{\text{BERT}}$ to compute the explanation alignment loss $\mathcal{L}_{\text{XAI}}$, which encourages the model to learn not only accurate predictions but also consistent explanation behavior during training.

3.2 Loss Function Definition

To align the target MLP model with the base model's reasoning process, we use a composite loss function that balances prediction accuracy and explanation consistency:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{CE}} + (1 - \alpha) \cdot \mathcal{L}_{\text{XAI}} \tag{1}$$

Here, $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss based on the ground-truth label, and $\mathcal{L}_{\text{XAI}}$ denotes the explanation alignment loss. The hyperparameter α controls the trade-off between accuracy and explanation consistency.

3.2.1 Prediction Loss $\mathcal{L}_{\text{CE}}$: Cross-Entropy Based on Ground-Truth Labels

To measure how well the MLP model predicts the correct class, we use the standard cross-entropy loss between the predicted softmax probabilities and the one-hot encoded ground-truth labels:

$$\mathcal{L}_{\text{CE}} = -\sum_{c=1}^{C} y_c \cdot \log(\hat{y}_{\text{MLP},c}) \tag{2}$$

where:

- C: number of classes,
- y_c: the ground-truth one-hot label vector,
- $\hat{y}_{\text{MLP},c}$: softmax output for class c from the MLP model.

This loss measures the accuracy of the prediction and is weighted by α in the total loss function.

3.2.2 Explanation Loss $\mathcal{L}_{\text{XAI}}$: Mean Squared Error-Based Explanation Alignment

In this study, we observe that the explanation vectors generated by BERT and MLP via Integrated Gradients (IG) have a multi-dimensional structure. To reflect this structure, we define our explanation loss $\mathcal{L}_{\text{XAI}}$ as the mean of class-wise mean squared errors (MSE) between the explanation vectors of the two models. For each class $c \in \{1, \ldots, C\}$, we extract the corresponding column vector

from both explanation matrices and compute the MSE between them. The final loss is obtained by averaging these values across all classes.

Let $E_{\text{BERT}}, E_{\text{MLP}} \in \mathbb{R}^{n \times C}$ denote the explanation matrices from the BERT and MLP models, respectively. The loss is defined as:

$$\mathcal{L}_{\text{XAI}} = \frac{1}{C} \sum_{c=1}^{C} \text{MSE}\left(E_{\text{BERT}}^{(:,c)},\ E_{\text{MLP}}^{(:,c)}\right)$$

where:

- $E^{(:,c)}$: the explanation vector (1D) for class c across all tokens,
- MSE: mean squared error.

This loss encourages the MLP model to align its token-level explanation patterns with those of BERT, ensuring similarity not only in prediction outcomes but also in the underlying reasoning process.

4 Experiments

4.1 Data

The training dataset for MLP model includes 10,000 samples from the AG News training dataset. The dataset is uniformly distributed across four classes, with 2,500 samples per class. Input texts for both BERT and MLP are tokenized by the same tokenizer,"BertTokenizerFast" [1], ensuring that the same tokens are processed by BERT and MLP from a sentence. The outputs of the MLP model are soft labels for each AG News class. The entire AG News test dataset is used for the validation of this experiment including 7,600 samples.

4.2 Evaluation method

We evaluate our method using two primary criteria: classification performance and resource efficiency. Accuracy is used to measure the model's ability to correctly classify input samples. To assess resource efficiency, we examine model size, number of parameters, and GPU memory usage, which are especially important in deployment scenarios with limited computational resources. As a baseline, we compare against a plain MLP model trained solely with cross-entropy loss $\mathcal{L}_{\text{CE}}$.

Evaluation metrics:

- **Accuracy**: measures classification performance on the test set.
- **Model size, number of parameters, and GPU memory usage**: measure computational efficiency.

4.3 Experimental details

We used a simple 3-layer MLP model that has 1 embedding layer and 2 fully-connected layers. Embedding layer tokenizes input text and assigns tokens to a 128-dimension vector, which is max number of tokens with padding. Then embedded vector is passed through 2 fully-connected layers with 256 hidden dimensions and 4 output dimensions. Loss function of this model is defined as in Section 3.2.

We use BERT as the teacher model. MLP model is trained for 5 epochs on the training dataset with a learning rate of 0.001. We evaluate the classification accuracy of our model on the test set while varying α from 0.0 to 1.0 in increments of 0.1. This α controls the weighting between cross-entropy loss and explanation loss, with α = 0.1 corresponding to a 1:9 ratio between the two.

4.4 Results

- Figure 2 shows the test accuracy of the MLP model across different α values. Notably, when trained without cross-entropy loss ($\alpha = 0.0$), the model still achieved 78.13

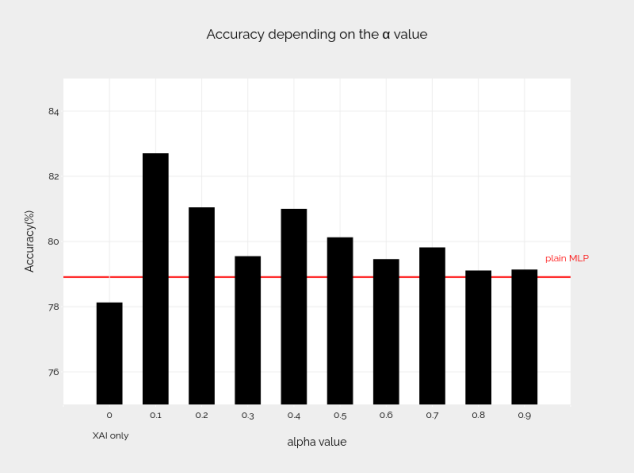

Figure 2: Test accuracy across different α values. The model achieves the highest accuracy (82.71%) when $\alpha = 0.1$, indicating an optimal balance between cross-entropy loss and explanation alignment.

Model	Accuracy	# of Parameters	Model Size	GPU Memory Usage
Plain MLP	78.91%	3.94M	15.04MB	321 MiB
MLP_XAI	82.71%	3.94M	15.04MB	321 MiB
BERT	94.51%	109M	417.68MB	1023 MiB

Table 1: Comparison of model efficiency and performance across architectures.

- The highest performance was observed at $\alpha = 0.1$, where the model reached 82.71
- Table 1 compares the performance of three models: the original BERT model, a plain MLP trained with cross-entropy only, and our MLP trained with the optimal $\alpha = 0.1$. Our model demonstrates a strong trade-off between performance and efficiency, significantly outperforming the plain MLP while remaining far more lightweight than BERT.

5 Analysis

First, our MLP model achieves its highest performance when $\alpha = 0.1$, indicating that a relatively small contribution from the cross-entropy loss (10%) combined with a dominant explanation loss (90%) is optimal. This suggests that the IG-based supervision provides a strong learning signal that effectively guides the MLP classifier.

Second, when the model is trained with $\alpha = 0$, it relies solely on the explanation loss $\mathcal{L}_{\text{XAI}}$, without using any cross-entropy supervision. Despite this, it achieves an accuracy of 78.13%, which is remarkably close to that of the plain MLP model trained with cross-entropy alone (78.91%). This observation highlights that even in the complete absence of label supervision, the explanation vectors derived from BERT's IG attributions contain enough information to guide the model toward meaningful classification behavior. It suggests that explanation based guidance can act as a indirect supervisory signal, capable of transferring not just predictions but also implicit structural understanding of the input.

Overall, these findings demonstrate the value of integrating explainability into the learning process not only for interpretability, but also for improving performance and enabling indirect supervised training setups.

6 Conclusion

In this work, we proposed a lightweight explanation-guided learning framework that trains an MLP classifier to align not only with prediction labels but also with token-level attribution patten derived

from a BERT teacher model. By combining cross-entropy loss with an explanation loss based on Integrated Gradients, our approach encourages the student model to learn both what to predict and how to reason about its predictions. Experimental results show that this method improves performance over a plain MLP model while maintaining computational efficiency. Notably, even in the absence of label supervision, the model trained solely on explanation signals exhibited meaningful classification behavior, suggesting that XAI explanations can serve as a form of indirect supervision. These findings highlight the potential of using explanations not just for interpretation, but also as a viable training signal for compact and interpretable models.

One limitation of our study is that the optimal value of α was identified through exhaustive search over fixed intervals between 0.0 and 1.0, without theoretical justification for its optimality. This approach is computationally inefficient and lacks adaptability. As future work, we aim to develop an adaptive algorithm that dynamically adjusts α during training, guided by statistical properties, to achieve more principled and efficient optimization.

References

[1] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and Jamie Brew. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, page 38–45. Association for Computational Linguistics, 2020.

[2] Sara Parchami-Araghi, Wenjia Xie, Tian Zhao, Jie Yang, and Yuncheng Li. Good teachers explain: Explanation-enhanced knowledge distillation. In *European Conference on Computer Vision (ECCV)*, 2024.

[3] Xinya Hu, Wei Zhang, Donghyun Kim, and William Wang. Large language models as attribution regularizers. `https://arxiv.org/abs/2503.01234`, 2025. arXiv preprint arXiv:2503.01234.

[4] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[6] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, pages 3319–3328. PMLR, 2017.

[7] Arnab Neelim Mazumder, Niall Lyons, Ashutosh Pandey, Avik Santra, and Tinoosh Mohsenin. Harnessing the power of explanations for incremental training: A LIME-based approach, 2023. arXiv preprint.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. arXiv preprint arXiv:1503.02531.

[9] Riccardo Cantini, Alessio Orsino, and Domenico Talia. Xai-driven knowledge distillation of large language models for efficient deployment on low-resource devices. *Journal of Big Data*, 11(63), 2024.