AttriT: Attribution for Distilled Vision Transformer

Ehsan Aghazadeh Yash Kamoji Karthik Ravichandran
{eaghazadeh, ykamoji, kravichandra}@umass.edu

Abstract

In this work, we introduce AttriT, a novel knowledge distillation method designed to optimize the training of compact vision transformers for deployment in resource-constrained environments. Our approach extends traditional knowledge distillation techniques by incorporating a unique loss term that accounts for the attribution of each image patch's importance to the final prediction. This additional loss term enhances the alignment between the larger teacher model and the smaller student model, enabling the latter to more effectively mimic the former's capabilities. We demonstrate the efficacy of AttriT through extensive experiments on benchmark datasets such as CIFAR-10 and CIFAR-100, where it outperforms existing state-of-the-art methods. Our results indicate that AttriT closes the performance gap with larger models, making it an ideal solution for implementation on edge devices. Github: decomp-vision-transformer-KD

1. Introduction

While recent advancements in vision transformers have demonstrated remarkable success in image processing tasks, their deployment in computationally constrained environments, such as edge devices, poses significant challenges due to their substantial model size and slower inference speeds. To address these limitations, knowledge distillation has emerged as a promising technique to transfer capabilities from a larger model to a more compact counterpart. Originally conceptualized by [11] for neural networks, this method allows for the harnessing of a larger model's capabilities without fully utilizing its capacity during operations. Applying this technique to vision transformers, the approach involves training a smaller, less resource-intensive model to emulate the behavior of a larger one without sacrificing performance. Notable implementations of this concept include the adaptation of the vision transformer architecture into more compact forms, such as [20] and [10], which maintain competitive performance while being significantly faster and smaller. These models integrate novel distillation strategies that involve multiple loss terms across the transformer's layers to refine performance.

Although these smaller models approximate the effectiveness of their larger counterparts, a performance gap often persists. This work aims to further refine the knowledge distillation process for vision transformers by incorporating an additional loss function based on the attribution of importance across different regions of an input image. Specifically, it explores the use of this new loss function in refining the distillation process, potentially narrowing the performance gap while maintaining computational efficiency. This study builds on the ideas presented by [15], applying their concepts of attributive loss to enhance alignment between the teacher (large vision transformer) and student (compact vision transformer) models.

Our contributions are as follows:

- We adopt the GlobEnc metric to the vision domain to identify the importance of each patch existing in an image.
- We incorporate the GlobEnc metric to have a novel term for distillation loss.
- By evaluating our method on several benchmarks, we show that the proposed loss term is a reliable alternative for common distillation objectives.

2. Background and Related Work

Currently, many SOTA (State-of-the-art) models in the NLP domain use a Transformer, which has inspired many researchers in Computer Vision to bring these transformer architectures to perform a variety of vision tasks. This paper [7] introduces the concept called Vision Transformers and starting from basic classification task, segmentation [19] to multimodal application, such as MiniGPT-V2 [4].

Following the usage of transformers in vision, the researcher started exploring the method used in the NLP domain for Computer Vision. In this context, some approaches proven to work in NLP can be experimented here. One such approach is exploring Attribution-Driven Knowledge Distillation [23]. This adds to the other three approaches we discussed before. This method uses three losses to compute the final loss: a) Attribution Distillation Loss, b) KL Divergence loss and c) cross-entropy loss. Another approach [23] deals with tokens and similar pa-

per which compress the image token using a method called Adaptive Token Sampling [8].

Moreover, there exist several approaches, including Decompx [], token-to-token interaction techniques as discussed in [9], and attention analysis methods outlined in [5], aimed at elucidating the decision-making process of a model. Some methods focus on analyzing both local and global contributions of input tokens [18]

When it comes to explainability, there are multiple level of granularity that we can look in to : 1) Local Explanation and 2) Global Explanation. Local Explanation covers the methods like Feature Attribution-Based Explanation, Attention-Based Explanation, Example-Based Explanations. The main Feature attribution-Based Explanation follows the technique called Perturbation-Based Explanation which work by perturbing input examples such as masking, altering, or/and removing input features and evaluating model output changes. The common strategy is leave-one-out that modify/removing the inputs tokens(Text) or patches (Image) at various levels that includes embedding vectors, hidden units [14], words, tokens and spans [24] to measure feature importance.

Other Attribution-Based Explanation methods includes Gradient-Based Explanation [17], Decomposition-Based Methods [2] and Surrogate Models [3]. When it comes to Attention-Based Explanation, Visualizations is most predominate method to understand the contribution of each tokens. Function-Based methods is another technique, since using only raw attention is not sufficient enough to explain model predictions completely. When comes to Image data, [8] this work introduces a parameter-free Adaptive Token Sampler (ATS) module for vision transformers, reducing computational costs by adaptively sampling significant tokens. Integrated as an additional layer along with attention layers. This seems to be useful in our work where, we not now take the attention based losses but also the ATS loss along with the attention.

Therefore, we propose to use the most recent attribution loss [16] to the self-attention layer of the DeiT-III backbone architecture [21] with a masked augmented [10] model. The intuition with the attribution loss is to give more importance to the image features among different layers in transformer that contribute the most in predicting the labels in the student model. We aim to make the student model more efficient in classifying the images. For further downstream tasks such as object segmentation, and VLMs (vision-language models) we can fine-tune this student model much faster without losing too much accuracy performance.

3. Methodology

3.1. Computing Attribution Scores

To compute attribution scores, we use the method proposed in [15] with several changes. To this end, we decompose components existing in the encoder block.

The self-attention mechanism Vaswani et al. [22] is the fundamental element of the encoder, responsible for the information mixture of a sequence of token representations $(x_1, \ldots, x_n)$. Each self-attention head computes a set of attention weights $A_h = \{\alpha_{h_{i,j}} | 1 \leq i, j \leq n\}$, where $\alpha_{h_{i,j}}$ is the raw attention weight from the i^{th} token to the j^{th} token in head $h \in \{1, \ldots, H\}$. Then we can write $z_i \in \mathbb{R}^d$, the output representation for the i^{th} token of a multi-head (with H heads), as below:

$$z_i = \text{CONCAT}(z_i^1, \ldots, z_i^H) W_O \tag{1}$$

$$z_i^h = \sum_{j=1}^{n} \alpha_{i,j}^h v^h(x_j) \tag{2}$$

where $v^h(x_j) = x_j W_v^h + b_v^h$.

Now by reformulating Equation 1, we can consider z_i as a summation over the attentions heads:

$$\begin{aligned} z_i &= \sum_{h=1}^{H} \sum_{j=1}^{n} \alpha_{i,j}^h v^h(x_j) W_O^h \\ &= \sum_{h=1}^{H} \sum_{j=1}^{n} \alpha_{i,j}^h \underbrace{v^h(x_j) W_O^h}_{f^h(x_j)} \end{aligned} \tag{3}$$

By merging W_O^h and W_v^h and discarding b_v^h, we have new z_i:

$$z_i = \sum_{j=1}^{n} \sum_{h=1}^{H} \alpha_{i,j}^h x_j W_{\text{Att}}^h \tag{4}$$

Our decomposition is expanded by incorporating layer normalization and residual connection components through the implementation of the encoder block:

$$z_i^+ = \sum_{j=1}^{n} \sum_{h=1}^{H} \alpha_{i,j}^h x_j W_{\text{Att}}^h + x_i \tag{5}$$

x_j and x_i can be written as $x_j = \sum_{k=1}^{N} x_{j \leftarrow k}$ and $x_i = \sum_{k=1}^{N} x_{i \leftarrow k}$, respectively. Now we have:

$$\begin{aligned} z_i^+ &= \sum_{h=1}^{H} \sum_{j=1}^{N} \alpha_{i,j}^h \sum_{k=1}^{N} x_{j \leftarrow k} W_{\text{Att}}^h + \sum_{k=1}^{N} x_{i \leftarrow k} \\ &= \sum_{k=1}^{N} \underbrace{\sum_{h=1}^{H} \sum_{j=1}^{N} \alpha_{i,j}^h x_{j \leftarrow k} W_{\text{Att}}^h + x_{i \leftarrow k}}_{z_{i \leftarrow k}^+} \end{aligned} \tag{6}$$

We now proceed with layer normalization:

$$\tilde{z}_i = \text{LN}(z_i^+) = \text{LN}\left(\sum_{k=1}^{N} z_{i\leftarrow k}\right) \tag{7}$$

Once more, in order to broaden the decomposition across the LN function, we utilize a method proposed by [12]. This technique involves breaking down the LN function into a summation of a new function denoted as $g(\cdot)$:

$$\text{LN}(z_i^+) = \sum_{k=1}^{N} \underbrace{g_{z_i^+}(z_{i\leftarrow k}) + \beta}_{\tilde{z}_{i\leftarrow k}} \tag{8}$$

$$g_{z_i^+}(z_{i\leftarrow k}^+) := \frac{z_{i\leftarrow k}^+ - m(z_{i\leftarrow k}^+)}{s(z_i^+)} \odot \gamma \tag{9}$$

where $m(\cdot)$ and $s(\cdot)$ correspond to the element-wise mean and standard deviation of the input vector.

By extending our decomposition, we add the layer normalization existing outside the attention block:

$$\tilde{z}_i^+ = \text{FFN}(\tilde{z}_i) + \tilde{z}_i \tag{10}$$

x_i is obtained by applying layer normalization to $\tilde{z}_i^+$:

$$\tilde{x}_i = \text{LN}(\tilde{z}_i^+) \tag{11}$$

Using the LN decomposition rule presented in Eq. 11, we separate the effects of residual and FFN output:

$$\tilde{x}_i = \sum_{k=1}^{n} \left(g_{\tilde{z}_i^+}(\text{FFN}(\tilde{z}_{i\leftarrow k})) + g_{\tilde{z}_i^+}(\tilde{z}_i^{\leftarrow k}) \right) + \beta \tag{12}$$

As a result, we obtain a representation for each token that illustrates its contribution within an encoder block:

$$\tilde{x}_{i\leftarrow k} \approx g_{\tilde{z}_i^+}(\tilde{z}_{i\leftarrow k}) = \frac{\tilde{z}_{i\leftarrow k} - m(\tilde{z}_{i\leftarrow k})}{s(\tilde{z}_i^+)} \tag{13}$$

We can now employ these representations to introduce a novel term in the loss function for knowledge distillation. We introduce A_t matrix used in 6.2 as below:

$$A_t := (||\tilde{x}_{i\leftarrow k}||) \in \mathbb{R}^{n\times n} \tag{14}$$

3.2. Setup

We prepare a baseline performance by fine-tuning the Teacher model on CIFAR 10 and CIFAR 100 [13] dataset. For our experiments, we used the Googles ViT-B/16 [7] (T) and Facebooks DeiT-distilled-B/16 (S) [21] model which are pre-trained on ImageNet [6].

To get the best performance for our distillation, we compare the accuracy's by fine-turning the teacher model and student model separately. This is will serve as an essential reference for our experiment analysis since we expect to get better accuracy's with distillation. This is shown in 1.

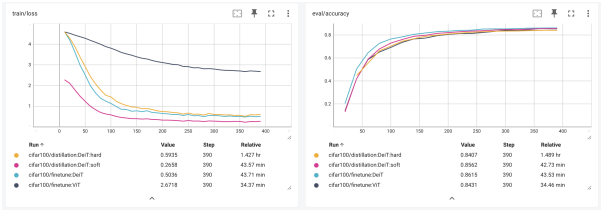


Figure 1. Baseline distillation comparison

We explore both types of distillation training of "soft" and "hard" distillation [21]. In soft distillation, the student model leverages "soft" labels from the teachers network which is the output vector of the softmax function rather than just the maximum of scores. We observe that the soft distillation improves the performance of the student model better.

We use this as the starting point for our experiments. The baseline distillation of our training consists of cross entropy loss of the student outputs and ground truth labels, as well as the KL divergence soft loss between the teacher and student output logits.

$$L_{CE} = CE(\psi(Z_s), y)$$

$$L_{KL} = (1-\lambda)CE(\psi(Z_s), y) + \lambda\tau^2 KL(\psi(\frac{Z_s}{\tau}), \psi(\frac{Z_t}{\tau}))$$

$$L_{model} = L_{CE} + L_{KL}$$

where Z_t are the logits of the teacher model, Z_s the logits of the student model, τ is the temperature for the distillation, λ is the coefficient balancing the Kullback–Leibler divergence loss (KL) and the cross-entropy (L_{CE}) on ground truth labels y, and ψ the softmax function.

3.3. Knowledge Distillation

Following the [21], we use knowledge distillation from the teacher T to student S, our model, to fine-tune S. In this approach, knowledge distillation happens at both the prediction layer and intermediate layers. We incorporate the following loss function.

$$L_{\text{model}} = L_{\text{CE}} + L_{\text{KL}} + L_{\text{Attr}} + L_{\text{Ats}}$$

We explain each term with details in what comes next.

Prediction Layer - L_{CE} and L_{KL}. Following [21], we use the cross-entropy (L_{CE}) on ground truth labels.

Intermediate Layers - L_{Attr}. We add additional loss term drawn from attribution scores:

$$L_{\text{Attr}} := \sum_{l=1}^{L} MSE(A_t^{l^T}, A_t^{l^S}) \tag{15}$$

Intermediate Layers - L_{Ats}. In addition to L_{Attr}, we also consider L_{Ats} in some of our experiments:

$$L_{\text{Ats}} := \sum_{l=1}^{L} MSE(N_{ATS}^{l^T}, N_{ATS}^{l^S}) \tag{16}$$

Where

$$N_{\text{ATS}} := \left(\frac{A_{i,j} \times \|\mathbf{V}_j\|}{\sum_{k=2}^{n} A_{i,k} \times \|\mathbf{V}_k\|} \right) \in \mathbb{R}^{n \times n} \tag{17}$$

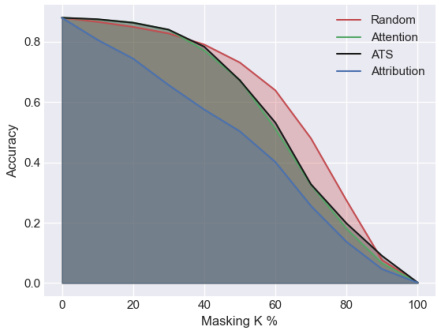


Figure 2. Area Under the Curve for Masking Accuracy

4. Experiments and Results

We conduct our analysis in diverse ways: 1) Evaluating score importance through visualizations, termed as faithfulness verification, 2) Examining different models on multiple datasets and exploring various loss combinations across models and datasets.

When it comes to Knowledge Distillation, the most prevalent approach is use Cross Entropy and KL losses. Additionally, we explored ATS [8] and Attribution [15] losses as well.

4.1. Faithfulness Verification

Faithfulness verification assesses whether a layer effectively captures important features of input tokens, represented by image patches. We compare three scores: Attribution, ATS, and Attention. By thresholding scores and masking important patches, we gauge the method's impact.

To visualize the feature scores, we use different strategies: 1) **Rollout** [1] that uses the representation of features by taking the dot product from every previous layer and accumulating the overall feature at the last layer of the transformer 2) **Plus** that uses summation of the feature across the layers 3) **SkipPlus** that only uses the last half layers.

From our experiments, we see the best representation for image classification is when plus is used. The feature importance can be seen in 3 . We can observe that attribution scores perform consistently, especially in cases when even attention doesn't consider the important regions of the images as important.

Masked images are predicted by the existing model, revealing how each method prioritizes features crucial to final predictions. Our experiments involve varying levels of masking percentages (0 to 90), providing insights into the efficacy of these approaches in capturing and utilizing essential information for accurate predictions. The masking strategy can be seen in this visualization 8.

We randomly selected three images from a dataset (Fish, Sparrow, and Graduation) and masked them based on ATS, Attention and Attribution scores and compared them with random masking. Then, we plot the prediction accuracy concerning mask percentage and masking methods. We expect the accuracy to reduce faster with methods that represent the features well. The AUC curve is shown in 2.

4.2. Distillation

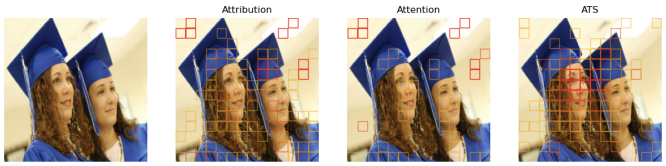
Our primary aim is to optimize the training of a student model by experimenting with various loss functions and selecting the most effective combination to facilitate efficient learning from the teacher model. We sequentially introduced different losses, evaluating their impact on prediction accuracy. Initially, using a ViT-B/16 model with Cross Entropy (CE) loss, we trained and assessed performance on CIFAR-10 and CIFAR-100 datasets. Focusing solely on final prediction probabilities for Knowledge Distillation, we employed a model without distillation tokens.

Subsequently, we incorporated KL (Hard or Soft) loss alongside CE in a model supporting distillation tokens (DeiT-B/16). We then integrated ATS with CE and KL, assessing accuracy. Similarly, we evaluated Attribution (ATR) loss in place of ATS, and finally combined ATS and ATR with CE and KL, training the model using these composite losses. The results of this training are shown 6 and 7.

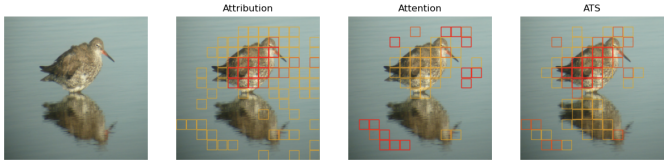
4.3. Results

We observe the best accuracy in both datasets when we use attribution loss during distillation training. Furthermore, we can see less accuracy in the same number of epochs when performing distillation with both ATS and Attribution loss are used. This analysis makes sense when considering that these two expressions are going against each other and performing distillation with either of them will give better results than combined.

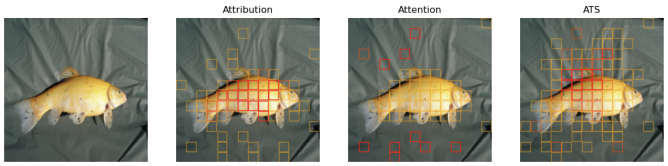
During the distillation training, figures 4 & 5, show the ATS loss is bounded in between a range while Attribution losses show a decline. This observation gives some insight into the student model behaviour for the two distillation training. Since there is a clear decline in the attribution loss, the student model learned more than to ATS loss. This lends credence to our results where we observe higher accuracy



(a) Graduation



(b) Redshank



(c) Tinca tinca

Figure 3. Patch scoring based on features

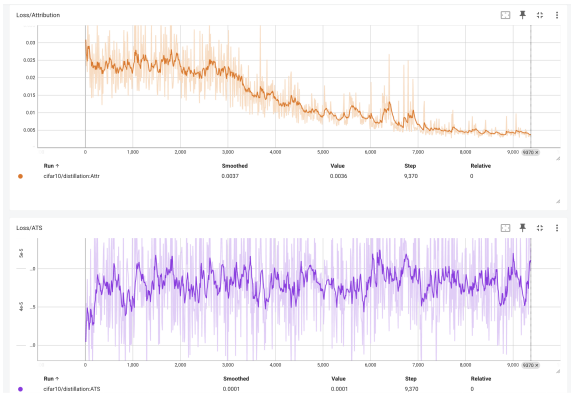


Figure 4. ATS vs Attribution loss for CIFAR 10

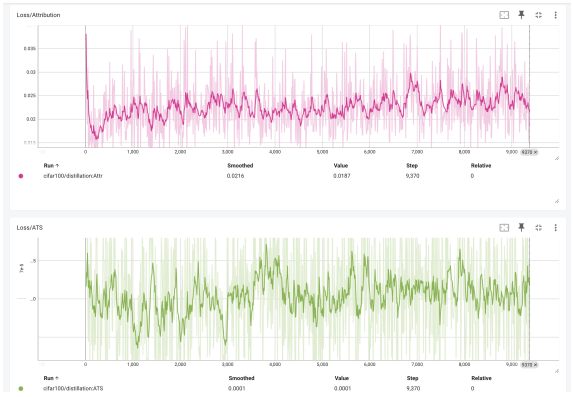


Figure 5. ATS vs Attribution loss for CIFAR 100

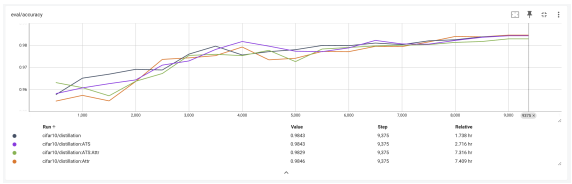


Figure 6. CIFAR 10 distillation training

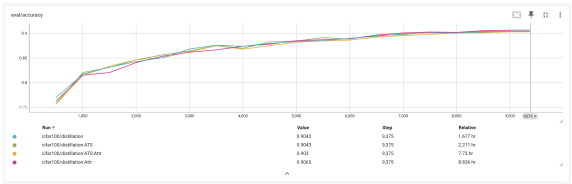


Figure 7. CIFAR 100 distillation training

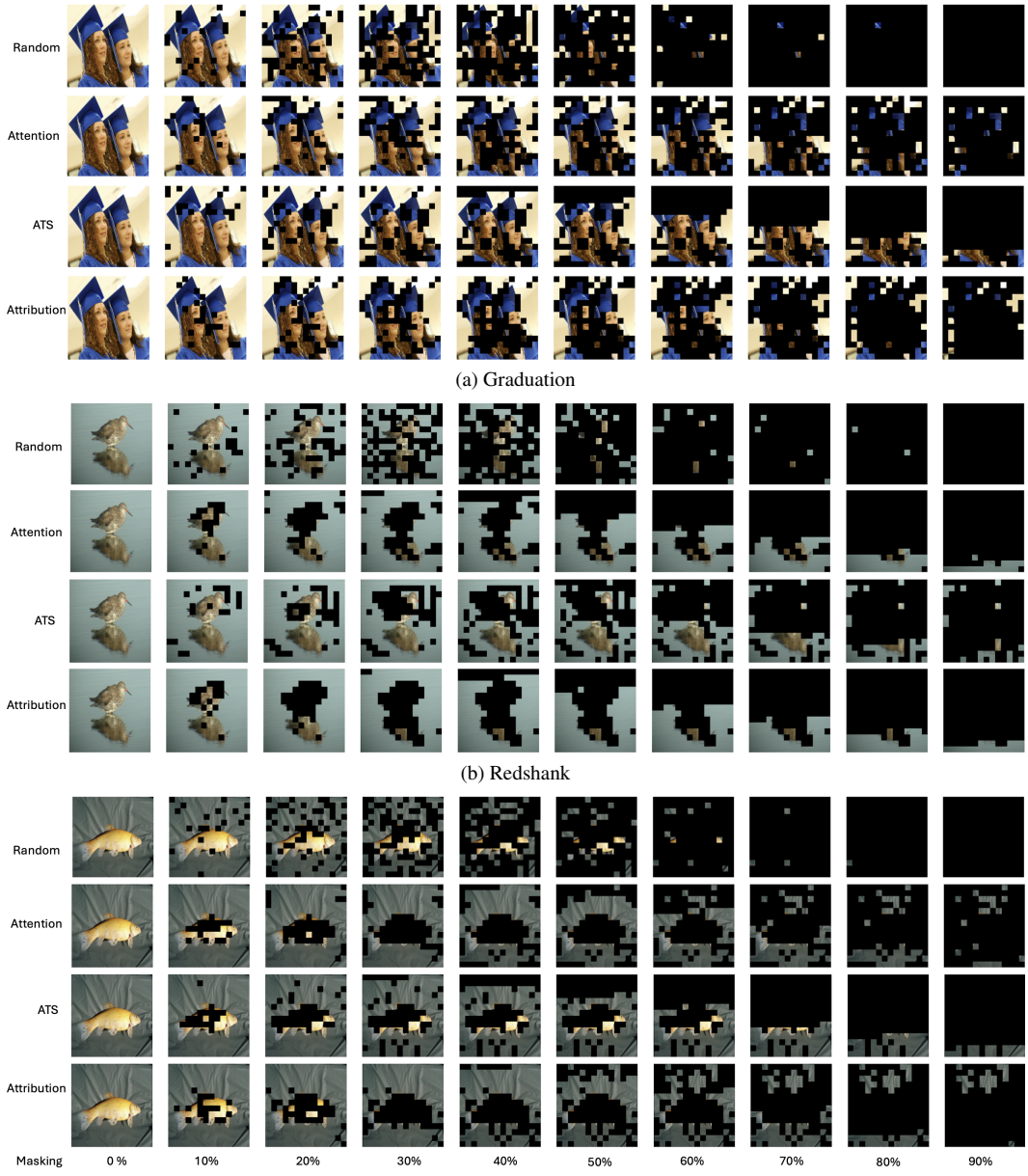
for attribution distillation.

Additionally, we can observe attribution distillation is very slow as compared to the other distillations. This is because the decomposed token calculation is computationally expensive. However, this can be optimized and the overall time can be brought down to comparable levels.

5. Conclusion

Table 1 shows the improvements we get on different distillation strategies. We see good accuracy improvements on both datasets. It is worth noting that the improvement for CIFAR 100 is more for CIFAR 10 even though CIFAR 100 has more labels for classification and hence a little harder.

We have shown that GlobEnc is a reliable metric for



(a) Graduation

(b) Redshank

(c) Tinca tinca

Figure 8. Patch masking based on various scores

measuring the contribution scores of each token to others. From the various analysis and expirements done here, we can see that leveraging GlobEnc during distillations produces additional information from the teacher model and distilled it into the student model. GlobEnc representations provides reliable feature importance, not only in the NLP domain but also for the classification tasks.

5.1. Future works

Due to limitations on computing resources, we didn't perform experiments on other datasets. However, we expect that the attribution distillation would be even more promising with bigger image sizes. With smaller images, region importance is imbalanced as there is less number of pixels that contribute towards the classification. From our experiments, we believe with more pixels (patches), there will be more information distilled from the teacher model to the student model.

Model	Loss	CIFAR 10	CIFAR 100
ViT-B/16	CE	98.21	89.34
DeiT-B/16	CE + KL	98.43(+0.22)	90.43(+1.09)
DeiT-B/16	CE + KL + ATS	98.43(+0.22)	90.43(+1.09)
DeiT-B/16	CE + KL + ATR	98.46**(+0.25)**	90.65**(+1.31)**
DeiT-B/16	CE + KL + ATS + ATR	98.29(+0.08)	90.30(+0.96)

Table 1. Knowledge Distillation with various Loss combinations

To have more comprehensive results, we will expand our experiments by evaluating our method on student models with less number of layers and smaller hidden vectors. By shrinking the model size, we can potentially run fast evaluations on edge devices that have fewer compute resources available.

Furthermore, our experiments were restricted to only image classification. However, there is much scope for various downstream tasks such as image segmentation, detection, facial recognition etc.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers, 2020. 4

[2] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, 2021. 2

[3] Hugh Chen, Ian C. Covert, Scott M. Lundberg, and Su-In Lee. Algorithms to estimate shapley value feature attributions, 2022. 2

[4] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. MiniGPT-v2: Large language model as a unified interface for vision-language multi-task learning. 2024. 1

[5] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, 2019. Association for Computational Linguistics. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 3

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 2021. 1, 3

[8] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Juergen Gall. Adaptive token sampling for efficient vision transformers, 2022. 2, 4

[9] Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8698–8714, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. 2

[10] Byeongho Heo, Taekyung Kim, Sangdoo Yun, and Dongyoon Han. Masking augmentation for supervised learning. 2024. 1, 2

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 1

[12] Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4547–4568, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 3

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 3

[14] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California, 2016. Association for Computational Linguistics. 2

[15] Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States, 2022. Association for Computational Linguistics. 1, 2, 4

[16] Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. Decompx: Explaining transformers decisions by propagating token decomposition. 2023. 2

[17] Hosein Mohebbi, Ali Modarressi, and Mohammad Taher Pilehvar. Exploring the role of bert token representations to explain sentence probing results, 2021. 2

[18] Damian Pascual, Gino Brunner, and Roger Wattenhofer. Telling BERT's full story: from local attention to global aggregation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 105–124, Online, 2021. Association for Computational Linguistics. 2

[19] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. 2024. 1

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1

[21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers distillation through attention. 2021. 2, 3

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2

[23] Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. Ad-kd: Attribution-driven knowledge distillation for language model compression. 2023. 1

[24] Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online, 2020. Association for Computational Linguistics. 2