LAYA: LAYER-WISE ATTENTION AGGREGATION FOR INTERPRETABLE DEPTH-AWARE NEURAL NETWORKS

A PREPRINT

Gennaro Vessio
Department of Computer Science
University of Bari Aldo Moro
Bari, Italy
gennaro.vessio@uniba.it

ABSTRACT

Deep neural networks typically rely on the representation produced by their final hidden layer to make predictions, implicitly assuming that this single vector fully captures the semantics encoded across all preceding transformations. However, intermediate layers contain rich and complementary information—ranging from low-level patterns to high-level abstractions—that is often discarded when the decision head depends solely on the last representation. This paper revisits the role of the output layer and introduces **LAYA** (*Layer-wise Attention Aggregator*), a novel output head that dynamically aggregates internal representations through attention. Instead of projecting only the deepest embedding, LAYA learns input-conditioned attention weights over layer-wise features, yielding an interpretable and architecture-agnostic mechanism for synthesizing predictions. Experiments on vision and language benchmarks show that LAYA consistently matches or improves the performance of standard output heads, with relative gains of up to about one percentage point in accuracy, while providing explicit layer-attribution scores that reveal how different abstraction levels contribute to each decision. Crucially, these interpretability signals emerge directly from the model's computation, without any external post hoc explanations. The code to reproduce LAYA is publicly available at: https://github.com/gvessio/LAYA.

Keywords Deep learning · Attention mechanisms · Neural architecture design · Representation learning · Interpretability

1 Introduction

Deep learning architectures have achieved outstanding success across computer vision, natural language processing, and multimodal tasks. Despite differences in modality and structure, most deep networks share a common principle: the final prediction is obtained from the representation learned by the *last hidden layer*. Formally, given an input x processed by a model with parameters θ, the forward pass produces a sequence of L hidden representations

$$h_1, h_2, \ldots, h_L = f_\theta(x),$$

where f_θ denotes the composition of the hidden layers. The final prediction is then obtained by applying a task-specific output head to the last representation:

$$\hat{y} = \phi\big(W h_L + b\big),$$

where $\phi(\cdot)$ denotes a task-specific activation function. This conventional design implicitly assumes that the last embedding h_L encapsulates all relevant semantic information extracted by the network.

However, empirical and theoretical studies suggest that different layers capture distinct and complementary aspects of the input: early layers encode local patterns, middle layers capture structural relations, and deeper layers represent abstract semantics (e.g., [Bau et al., 2017, Tenney et al., 2019]). Popular architectures such as ResNet [He et al., 2016] and DenseNet [Huang et al., 2017] partially address this issue through skip or dense connections, facilitating gradient flow and feature reuse. Nevertheless, these mechanisms operate within the feature extractor, while the output head remains a static projection over the final representation. Similarly, multi-scale fusion models such as FPN [Lin et al.,

2017] and Transformer-based architectures [Vaswani et al., 2017, Dosovitskiy, 2020] exploit hierarchical information during feature learning but still collapse the entire representational hierarchy before classification.

This work revisits the output stage of deep neural networks. The central idea is that the output layer should not depend exclusively on the deepest representation but instead integrate information across depth. To this end, **LAYA** (*Layer-wise Attention Aggregator*) is introduced as a mechanism that learns to weight and combine all hidden representations $\{h_i\}_{i=1}^{L}$ in an input-dependent manner.

Each hidden state is mapped into a shared latent space through a lightweight adapter $g_i(\cdot)$, ensuring comparability across layers. LAYA then replaces the standard classifier head with an attention-based aggregator:

$$h_{\text{agg}} = \sum_{i=1}^{L} \alpha_i(x)\, g_i(h_i),$$

where $\alpha_i(x)$ are learnable, input-conditioned coefficients reflecting the relevance of each layer for the current sample. The final prediction is computed as:

$$\hat{y} = \phi(W h_{\text{agg}} + b).$$

This formulation transforms the output layer from a passive projection into an active module that reasons over the network's internal states. Crucially, the attention weights $\alpha_i(x)$ provide an explicit and quantitative attribution signal, revealing which layers influence the prediction. This yields interpretability that emerges directly from the model's computation, without relying on external post hoc techniques.

In summary, this work makes three main contributions. First, a core limitation of standard neural architectures is examined: the final prediction relies exclusively on the last hidden representation, disregarding the structured hierarchy of intermediate features learned throughout the network. Second, LAYA is introduced as a lightweight, architecture-agnostic output module that applies input-conditioned attention across all hidden layers, enabling depth-aware aggregation while leaving the backbone and its computational pipeline unchanged. Third, a systematic experimental framework is presented across both vision and language tasks, evaluating LAYA against competitive baselines and assessing the interpretability afforded by its layer-wise attention scores.

The rest of the paper is structured as follows. Section 2 surveys related work. Section 3 presents the LAYA mechanism and its integration into standard architectures. Section 4 describes the experimental setup, benchmarking results, and interpretability analyses. Section 5 discusses broader implications and potential directions for future work.

2 Related Work

Hierarchical representation learning. Deep neural networks organize computation hierarchically, with early layers capturing low-level features and deeper ones encoding progressively abstract semantics. Early convolutional models such as LeNet [LeCun et al., 2002], AlexNet [Krizhevsky et al., 2012], and VGG [Simonyan and Zisserman, 2014] demonstrated the effectiveness of deep hierarchies but suffered from vanishing gradients and feature degradation. Residual Networks (ResNet) [He et al., 2016] introduced skip connections to facilitate gradient flow and feature reuse, enabling the training of substantially deeper architectures. Densely Connected Networks (DenseNet) [Huang et al., 2017] further extended this principle by concatenating outputs from all preceding layers to encourage feature reuse and reduce redundancy. While these designs improve representational richness and allow information to flow across layers, the final classifier still receives a *single, collapsed representation* h_L. Although h_L is a function of all preceding transformations, the output layer cannot access intermediate features explicitly nor modulate their contribution based on the input; all depth-specific information is irreversibly compressed into the last embedding.

Multi-scale and feature fusion models. Many architectures explicitly integrate features across multiple spatial scales or abstraction levels. Feature Pyramid Networks (FPN) [Lin et al., 2017] and BiFPN [Tan et al., 2020] merge hierarchical features to enhance object detection and segmentation, while HRNet [Sun et al., 2019], U-Net++ [Zhou et al., 2018], and DeepLab [Chen et al., 2018] leverage encoder–decoder symmetries to preserve both spatial resolution and semantic depth. Vision Transformers (ViT) [Dosovitskiy, 2020] and hierarchical variants such as Swin Transformer [Liu et al., 2021] similarly exploit multi-stage attention. CrossViT [Chen et al., 2021] extends this direction by allowing cross-attention between representations extracted at different patch granularities. Cross-layer attention designs [Huang et al., 2022] further highlight that intermediate representations carry complementary task-relevant signals. More recently, LayerFusion [Zheng et al., 2024] demonstrated that aggregating intermediate representations can yield more contextual embeddings, though this fusion typically occurs *within* the encoder. In all these cases, hierarchical information is successfully incorporated during feature learning; however, the classifier still collapses the resulting hierarchy into a single representation before prediction, with no mechanism for input-dependent weighting across layers.

Output-level reasoning and interpretability. Despite substantial progress in feature fusion, the output layer in most architectures remains a static linear projection. Early-exit models such as BranchyNet [Teerapittayanon et al., 2016] attach auxiliary classifiers to intermediate layers for efficiency, and deep supervision approaches [Lee et al., 2015] encourage multiple layers to contribute training signals. However, these methods do not aggregate features at inference time. Ensemble strategies [Lakshminarayanan et al., 2017] combine multiple predictors, yet they require multiple forward passes. ScalarMix [Peters et al., 2018], used in contextual embeddings such as ELMo, mixes representations from different layers, but the mixing weights are global parameters shared across all inputs, limiting adaptability. Attention mechanisms such as Bahdanau attention [Bahdanau et al., 2014] learn relevance scores dynamically, but are typically applied across sequence or token dimensions rather than across network depth.

Positioning of LAYA. LAYA differs from prior approaches by introducing *output-level depth reasoning*: it operates strictly at the prediction stage and assigns *input-conditioned* attention weights to representations from all layers. Instead of relying on a single terminal embedding, LAYA aggregates $\{h_i\}_{i=1}^{L}$ through learned coefficients $\alpha_i(x)$, making the final decision explicitly dependent on multiple abstraction levels. This yields intrinsic interpretability, as the attention weights function as layer-attribution scores indicating which depths contribute most to each prediction. LAYA is architecture-agnostic, adds minimal parameter overhead, and requires no modification to the backbone or its training pipeline. Compared to ScalarMix-style approaches, which learn a *global*, input-independent mixture of layer representations, LAYA produces *per-sample* relevance weights and therefore adapts the depth contribution dynamically to each input. In contrast to cross-layer fusion mechanisms integrated within the encoder—such as LayerFusion or cross-layer attention—LAYA does not modify the backbone or alter feature learning: it acts as a *pure output head* that reasons over already-computed internal states. This design makes LAYA easy to retrofit onto existing architectures, even frozen ones.

3 Method

Motivation. Deep networks learn hierarchical representations, with different layers capturing increasingly abstract information. However, the standard output head relies exclusively on the final embedding h_L, discarding useful signals computed at earlier depths. LAYA (*Layer-wise Attention Aggregator*) addresses this limitation by allowing the prediction stage to integrate information from all layers in an input-dependent manner.

Overview. LAYA redefines the standard output head as a depth-aware aggregation module. Instead of relying solely on h_L, LAYA assigns input-dependent attention weights to all hidden representations $\{h_i\}_{i=1}^{L}$ and forms a weighted combination prior to prediction. This enables the model to adaptively emphasize different abstraction levels based on the input.

Formulation. Consider a backbone network with L hidden layers:

$$h_i = f_i(h_{i-1}), \quad i = 1, \ldots, L, \qquad h_0 = x.$$

In standard architectures, the prediction is computed as:

$$\hat{y} = \phi(W h_L + b).$$

LAYA generalizes this by treating all $\{h_i\}$ as relevant signals. Each h_i is first projected into a common latent space of dimension d^* by a learnable adapter:

$$g_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d^*}, \qquad z_i = g_i(h_i).$$

Then, an input-conditioned aggregation is computed as:

$$h_{\text{agg}} = \sum_{i=1}^{L} \alpha_i(x)\, z_i,$$

where the coefficients $\alpha_i(x)$ are attention weights specific to each input sample. The final prediction is obtained as:

$$\hat{y} = \phi(W h_{\text{agg}} + b).$$

Attention mechanism. Unlike standard attention mechanisms that operate over spatial or temporal dimensions, LAYA performs attention over the depth dimension of the network. The attention coefficients $\alpha_i(x)$ are generated by a small meta-network operating on the enriched representations of all layers. Each z_i can be optionally transformed by a shared mapping:

$$u_i = \psi(z_i),$$

where ψ is either the identity function or a lightweight two-layer MLP. The identity choice keeps the attention mechanism strictly linear with respect to the enriched features, while the MLP introduces an additional non-linearity. Both variants are evaluated in the ablation study. The resulting features $\{u_i\}$ are concatenated and passed to a scoring MLP that outputs unnormalized relevance logits:

$$s(x) = \mathrm{MLP}_{\text{score}}([\, u_1, u_2, \ldots, u_L \,]) \in \mathbb{R}^L.$$

These logits are normalized by a temperature-scaled softmax:

$$\alpha_i(x) = \frac{\exp(s_i(x)/\tau)}{\sum_{j=1}^{L} \exp(s_j(x)/\tau)},$$

where the temperature $\tau > 0$ controls the sharpness of the distribution (lower values encourage sparser, more decisive attention). Finally, the aggregated embedding is obtained as:

$$h_{\text{agg}} = \sum_{i=1}^{L} \alpha_i(x)\, g_i(h_i),$$

and passed to the output head for prediction.

The coefficients $\alpha_i(x)$ serve as explicit, per-input relevance scores over layers, providing a direct measure of how the model distributes representational importance across depth. Because these coefficients are learned jointly with the model, interpretability emerges as an *intrinsic* property of the architecture rather than an external post hoc explanation such as LIME [Ribeiro et al., 2016], SHAP [Lundberg and Lee, 2017], or Grad-CAM [Selvaraju et al., 2017].

Training and integration. LAYA is trained jointly with the backbone using the same objective (e.g., cross-entropy), without requiring auxiliary losses or multi-stage training procedures. It introduces minimal overhead: the projection adapters require $O(\sum_{i=1}^{L} d_i d^*)$ parameters, and the scoring network adds $O(Ld^{*2})$ parameters in the worst case, which remains modest for the values of L and d^* considered in the experiments. Since LAYA operates entirely at the output stage, the backbone requires no modification, and the module can be used as a drop-in replacement for the standard output head.

4 Experiments

LAYA was evaluated as a drop-in output head across both vision and language tasks, using three standard benchmarks and heterogeneous backbone architectures. The experimental setup was designed to (*i*) isolate the contribution of *depth-aware* aggregation at the output stage, (*ii*) measure performance differences against competitive baselines, and (*iii*) examine the interpretability signals exposed by the learned layer-attention weights. In addition, the interpretability analysis was extended to an artwork classification dataset, where the hierarchical nature of visual features is particularly pronounced; this setting allows assessing whether LAYA assigns greater importance to mid-level layers, which capture brushstroke- and texture-level information.

4.1 Experimental Setup

Datasets and backbones. The empirical evaluation spanned three popular and widely used benchmarks—Fashion-MNIST, CIFAR-10, and IMDB—covering low- to medium-complexity visual tasks and text classification. Although these datasets are relatively lightweight compared to large-scale industrial benchmarks, they are particularly well-suited for the scope of this work. LAYA operates strictly at the *output stage*, and its contribution must therefore be assessed *independently* of confounding factors such as large-scale pretraining, data augmentation pipelines, or high-capacity architectures. Using controlled, well-understood datasets enables isolating the effect of depth-aware aggregation while keeping the backbone fixed across methods. Moreover, these datasets span different modalities and representation hierarchies, providing a sufficiently diverse testbed for validating LAYA's generality without conflating improvements with domain-specific heuristics or model scaling. An additional interpretability-focused experiment on an artwork classification dataset further illustrates LAYA's behavior in domains where earlier visual features (e.g., texture, brushstroke) play a critical role, highlighting its ability to shift attention across depth levels based on the task.

Fashion-MNIST [Xiao et al., 2017] contains $60{,}000$ training and $10{,}000$ test grayscale images across 10 clothing categories at 28×28 resolution. For this dataset, a three-layer MLP with hidden sizes $[512, 256, 128]$ was used, with each layer followed by LayerNorm [Ba et al., 2016] and GELU activation [Hendrycks, 2016]. All intermediate activations were made available to LAYA.

CIFAR-10 [Krizhevsky and Hinton, 2009] comprises $50{,}000$ training and $10{,}000$ test color images in 10 object classes at 32×32 resolution. A lightweight CNN backbone was adopted, consisting of three sequential depthwise–pointwise convolutional stages [Howard et al., 2017], each followed by LayerNorm and GELU, with spatial downsampling in the latter stages. Global average pooled representations from each stage served as depth-specific features for LAYA.

For IMDB sentiment analysis [Maas et al., 2011], which includes $25{,}000$ labeled movie reviews (positive/negative), text was processed using a trainable embedding layer followed by stacked bidirectional LSTMs [Hochreiter and Schmidhuber, 1997]. The final hidden state of each BiLSTM layer was used as its corresponding layer representation.

To probe layer-wise interpretability in settings where perceptual hierarchies are especially prominent, an artwork classification dataset was also considered. Specifically, the *Best Artworks of All Time* collection[1] was used, which contains digitized paintings by a diverse set of artists, each with metadata including artistic period, biography, and stylistic descriptors. Only the images were employed, and the data were reorganized into a multi-class classification task aimed at capturing broad stylistic patterns rather than author identification. For this experiment, a pretrained ViT-Base/16 backbone [Dosovitskiy, 2020] (initialized from ImageNet-21k weights) was adopted and hidden states from all transformer encoder blocks were exposed to LAYA. This configuration provides a suitable testbed for assessing whether the aggregation mechanism naturally reallocates attention toward shallower or mid-level layers when fine-grained perceptual cues—such as brushstroke texture or local contrast patterns—play a decisive role. Moreover, it demonstrates the broader applicability of LAYA to more complex, pretrained architectures, indicating that the proposed output-level aggregation can be seamlessly integrated across a wide range of tasks and model families.

Baselines. To contextualize the contribution of depth-aware aggregation, three standard output heads were considered, differing only in how they combine the layer-wise representations $\{h_i\}_{i=1}^{L}$, while all backbones and training settings were kept identical:

1. LASTLAYER. Only the deepest representation is used for classification:

$$\hat{y} = \phi(W h_L + b).$$

   This corresponds to the conventional output design adopted in most deep architectures.

2. CONCAT. Each layer representation is first mapped to a shared dimensionality via a learnable adapter $g_i(\cdot)$; the resulting vectors are concatenated and passed through a small MLP prior to classification:

$$h_{\text{agg}} = \text{MLP}_{\text{post}}\big([\, g_1(h_1),\, \ldots,\, g_L(h_L)\,]\big), \qquad \hat{y} = \phi(W h_{\text{agg}} + b).$$

   This produces a uniform, input-independent fusion of multi-depth features.

3. SCALARMIX [Peters et al., 2018]. A set of learnable scalar weights determines each layer's overall contribution on average across the dataset:

$$h_{\text{agg}} = \sum_{i=1}^{L} \alpha_i\, g_i(h_i), \qquad \alpha_i = \frac{e^{s_i}}{\sum_{j=1}^{L} e^{s_j}}, \quad s_i \in \mathbb{R}.$$

   SCALARMIX is a widely adopted baseline in NLP, particularly in contextual representations such as ELMo, where it effectively balances information across depth. It preserves layer identities and performs multi-layer fusion, making it the closest baseline to LAYA conceptually; however, the weighting is *global* and does not adapt to individual inputs.

Training details. All models were implemented in TensorFlow/Keras and trained with mini-batch optimization. The optimizer was Adam [Kingma and Ba, 2015] with learning rate in the range 3×10^{-4}–10^{-3}, cross-entropy as the loss, accuracy as the primary metric, and a fixed 10% validation split. Early stopping on validation accuracy with patience 5 was applied, restoring the best checkpoint, and training was capped at 50 epochs. Experiments were repeated over five fixed seeds, and results are reported as mean $\pm$ standard deviation across runs. Inputs were normalized to $[0, 1]$ and flattened for Fashion-MNIST, channel-wise normalized for CIFAR-10, and tokenized to a 20k vocabulary for IMDB.

The LAYA head was tuned by grid search independently for each dataset. For Fashion-MNIST, the search spanned $d^* \in \{64, 96, 128\}$, $\tau \in \{0.5, 1.0, 1.5\}$, $\psi \in \{\texttt{identity}, \texttt{mlp}\}$, and MLP scorer width in $\{d^*, 2d^*\}$. For CIFAR-10, the same settings were used except for $d^* \in \{128, 192, 256\}$. For IMDB, the search considered $d^* \in \{64, 96\}$ and $\tau \in \{0.7, 1.0, 1.3\}$ with the same choices for ψ and scorer width. Each configuration was evaluated across three seeds, and the best configuration was selected based on the highest mean validation accuracy. The CONCAT and SCALARMIX baselines underwent parallel ablations over d^* to ensure comparability.

[1] `https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time`

Following hyperparameter tuning, all heads (LASTLAYER, CONCAT, SCALARMIX, and LAYA) were trained under identical settings and evaluated over the full five-seed protocol. Alongside accuracy, macro-F1 and two-sided 95% confidence intervals were computed using t-statistics. To support interpretability analysis, attention weights α were also extracted.

All models shared the same backbone architecture, allowing the effect of the output mechanism to be isolated. As with any parametric model, the adequate capacity of LAYA depended on the intrinsic complexity of each dataset; tuning, therefore, corresponded to evaluating the method in the regime in which it was intended to operate, rather than to additional optimization. Baselines were sized according to the same principle.

For the artwork classification experiment, as mentioned earlier, a pretrained ViT-Base/16 backbone [Dosovitskiy, 2020] (initialized from ImageNet-21k weights) was coupled with a LAYA head operating on the CLS token of all encoder blocks. The backbone parameters were kept frozen, so that only the LAYA aggregation module and the final classifier were optimized, allowing the analysis to focus on depth-aware aggregation rather than on representation learning. The *Best Artworks of All Time* images were split into training, validation, and test sets, with an 80/10/10 stratified split across style labels. All images were resized to 224×224 and normalized to $[0, 1]$. Optimization was carried out with the Adam optimizer, a learning rate of 3×10^{-5}, and early stopping on validation accuracy with patience 3, for a maximum of 15 epochs. No additional data augmentation was applied to preserve the stylistic characteristics of each artwork. This experiment was conducted with a single ViT+LAYA configuration and was used solely to study layer-wise attention profiles in a setting where fine-grained visual style plays a central role, rather than to benchmark alternative heads.

4.2 Quantitative Results

Benchmarking. The quantitative comparison across the three benchmarks is reported in Table 1. In all cases, the backbone architecture was held fixed and only the output aggregation head was varied. Across Fashion-MNIST, CIFAR-10, and IMDB, LAYA consistently achieved the highest mean accuracy and macro-F1 scores. On Fashion-MNIST, LAYA reached 0.8834 ± 0.0046 accuracy, slightly exceeding LASTLAYER (0.8829 ± 0.0063) and outperforming both SCALARMIX and CONCAT. On CIFAR-10, LAYA obtained 0.7212 ± 0.0060 accuracy, marginally higher than CONCAT (0.7155 ± 0.0096), SCALARMIX (0.7144 ± 0.0171), and LASTLAYER (0.7112 ± 0.0236). A similar pattern was observed on IMDB, where LAYA reached 0.8707 ± 0.0035, slightly but consistently surpassing all baselines.

Notably, the 95% confidence intervals indicate that the improvements were stable across seeds. Since no data augmentation or task-specific optimization was applied and all methods were trained under identical protocols, these differences can be attributed directly to the aggregation mechanism rather than to variations in model capacity, preprocessing, or optimization schedules. The gains, therefore, reflect the advantage of allowing the output head to dynamically reweight intermediate representations instead of committing solely to the deepest one.

Two systematic trends emerged across datasets. First, although the improvements were generally modest, they tended to be slightly more noticeable on CIFAR-10, a dataset that benefits from multi-level abstraction but remains challenging for small custom models without augmentation. Second, LAYA exhibited consistently lower variance across random seeds, indicating improved stability in the resulting solutions. Both observations support the hypothesis that depth-aware aggregation provides a robust, adaptable representation at the output stage, even though the absolute differences in accuracy are small.

Ablation study. Table 2 reports the best-performing LAYA configurations identified for each dataset during the grid search tuning procedure described above. The projection dimension d^* and the attention temperature τ varied moderately across datasets, whereas the choice of the transformation $\psi(\cdot)$ affected only IMDB, where a small MLP yielded a measurable improvement. The scorer width followed the scaling of d^* but did not require large values. These results indicate that LAYA is not overly sensitive to architectural hyperparameters and achieves its gains through depth-aware aggregation rather than increased model capacity.

4.3 Interpretability

Beyond aggregate metrics, the analysis also considered the interpretability signals encoded in LAYA's layer-wise attention weights. Performance, in fact, is not the primary benefit of LAYA: its layer-wise attention scores provide an intrinsic explanation of how different abstraction levels support each prediction, a capability absent from standard output heads. For every input sample x, LAYA outputs both a class-probability vector and a set of attention coefficients $\alpha(x) \in \mathbb{R}^L$, which quantify the relative contribution of each layer to the final prediction.

Global layer-wise attention profiles. To analyze how LAYA distributes representational importance across depth, global statistics of the attention coefficients $\alpha_i(x)$ were computed for each hidden layer i on the full test sets of

Dataset	Head	Accuracy	Macro-F1	Acc. 95% CI
Fashion-MNIST	LASTLAYER	0.8829 ± 0.0063	0.8826 ± 0.0066	[0.8752, 0.8907]
	CONCAT	0.8789 ± 0.0020	0.8783 ± 0.0020	[0.8764, 0.8814]
	SCALARMIX	0.8807 ± 0.0026	0.8800 ± 0.0027	[0.8775, 0.8839]
	LAYA	0.8834 ± 0.0046	0.8827 ± 0.0052	[0.8777, 0.8891]
CIFAR-10	LASTLAYER	0.7112 ± 0.0236	0.7108 ± 0.0228	[0.6819, 0.7405]
	CONCAT	0.7155 ± 0.0096	0.7136 ± 0.0117	[0.7036, 0.7274]
	SCALARMIX	0.7144 ± 0.0171	0.7123 ± 0.0177	[0.6932, 0.7356]
	LAYA	0.7212 ± 0.0060	0.7205 ± 0.0053	[0.7138, 0.7286]
IMDB	LASTLAYER	0.8692 ± 0.0057	0.8691 ± 0.0057	[0.8621, 0.8762]
	CONCAT	0.8688 ± 0.0062	0.8687 ± 0.0063	[0.8611, 0.8765]
	SCALARMIX	0.8669 ± 0.0043	0.8669 ± 0.0044	[0.8614, 0.8723]
	LAYA	0.8707 ± 0.0035	0.8707 ± 0.0035	[0.8663, 0.8751]

Table 1: Quantitative comparison of output heads across datasets (5 runs per configuration). All models shared the same backbone and were trained under identical conditions. No data augmentation or task-specific optimization was applied.

Dataset	d^*	τ	$\psi(\cdot)$	Scorer Width
Fashion-MNIST	96	1.5	`identity`	192
CIFAR-10	128	1.0	`identity`	128
IMDB	96	1.3	`mlp`	192

Table 2: Best-performing LAYA configurations selected by grid search on each dataset.

Fashion-MNIST, CIFAR-10, and IMDB. For every dataset, the mean and standard deviation of $\alpha_i(x)$ were calculated, providing a compact summary of the average contribution of each layer to the final prediction (Fig. 1). The resulting profiles show that attention does not collapse to a uniform distribution: LAYA exhibits clear, dataset-specific depth preferences. On Fashion-MNIST, attention is primarily concentrated on the deepest layer (mean ≈ 0.76), although shallower layers still display notable variability, indicating that different samples rely on different depths even in a relatively simple visual task. On CIFAR-10, the distribution becomes markedly sharper, with the final layer dominating almost entirely (mean ≈ 0.96, with low variance), reflecting the higher representational demands of natural-image classification. The IMDB sentiment classifier shows a contrasting pattern: attention is evenly split between the two recurrent layers (means ≈ 0.48 and ≈ 0.52, both with slight variance), suggesting that the BiLSTM stack preserves complementary information across depths and that both layers contribute consistently to the decision.

These results indicate that LAYA adapts its depth usage in a strongly task-dependent manner: concentrated on deeper representations in complex vision tasks, more distributed in simpler visual settings, and balanced in text. This behavior confirms that LAYA does not merely aggregate layers but actively reweights representational depth in alignment with the structure and requirements of each domain.

More importantly, the global attention statistics provide a direct diagnostic tool for understanding how representational importance is allocated across the network. By quantifying the average contribution of each layer, LAYA enables identification of depth levels that consistently carry little predictive weight, detection of shifts in depth usage across tasks or modalities, and characterization of how architectural components participate in the decision process. Such information would otherwise require external probing methods or post hoc explanations.

Class-wise attention profiles. To investigate whether different semantic categories relied on different abstraction levels, class-wise attention profiles were computed for each dataset. For every class c and layer i, attention coefficients were averaged over all test samples with true label c,

$$\bar{\alpha}_{c,i} = \mathbb{E}[\alpha_i(x) \mid y = c],$$

and the analysis was further stratified into *all*, *correct*, and *incorrect* predictions. The resulting heatmaps (Fig. 2) provide a compact view of how LAYA distributes importance across depth for each class and allow a direct comparison across the three benchmarks.

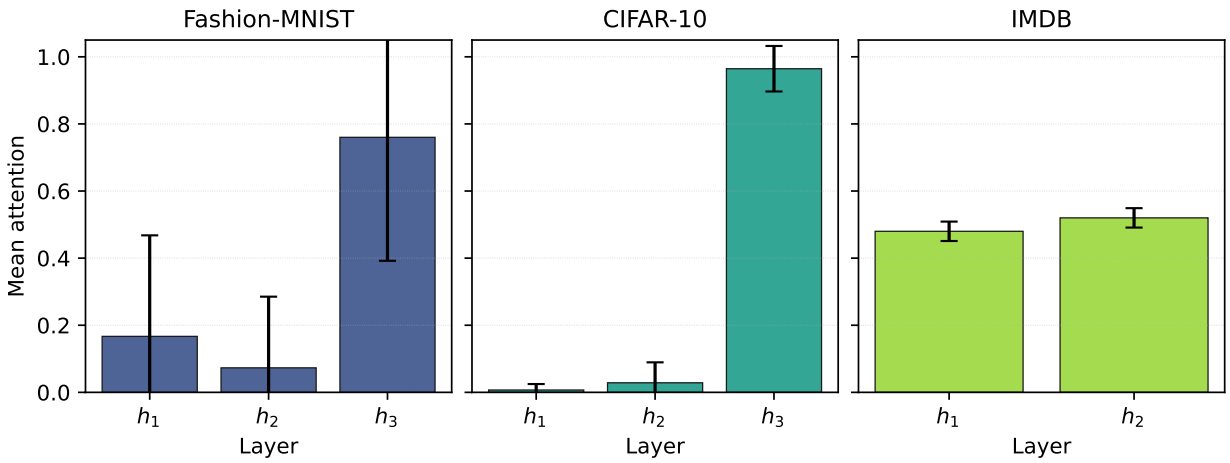

Figure 1: Global layer-wise attention statistics for LAYA (mean $\pm$ standard deviation over the full test set). LAYA consistently develops task-specific depth preferences: strong bias toward deep layers for CIFAR-10, softer depth usage for Fashion-MNIST, and nearly symmetric contributions in the IMDB text model.

On Fashion-MNIST, attention profiles are clearly class-dependent and markedly non-uniform. Most categories assign the largest share of attention to the deepest layer, but some systematically depart from this trend. For instance, class 5 (Sandal) concentrates most of its attention on the first layer in both the overall and correct subsets (mean ≈ 0.88 on h_1 for correct samples), while misclassified sandals shift part of the mass towards the deepest layer. A similar pattern is visible for class 0 (T-shirt/top): correct predictions favour the intermediate layer h_2, whereas incorrect ones move attention towards h_3. For class 9 (Ankle boot), both correct and incorrect predictions strongly rely on the last layer, but errors are associated with an even more extreme focus on h_3 and a reduced contribution from h_1. In contrast, categories such as Trouser (1), Pullover (2), Coat (4), Sneaker (7), and Bag (8) consistently prioritise the deepest layer, with only minor differences between correct and incorrect subsets. These patterns indicate that LAYA learns class-specific "attention fingerprints" over depth, and that shifts in where attention is concentrated correlate with success or failure on particular categories.

On CIFAR-10, attention becomes even more skewed: virtually all classes, irrespective of correctness, place almost all their mass on the deepest layer, with the first two layers receiving only residual weights. Differences between correct and incorrect predictions mainly translate into small redistributions among the earlier layers, confirming that high-level convolutional features dominate discrimination in this setting.

On IMDB, the behaviour is qualitatively different. With only two recurrent layers, attention remains nearly balanced between them for both sentiment classes and across correctness subsets, with mean values oscillating around 0.5 for h_1 and h_2. The heatmaps do not reveal strong class-specific signatures: the BiLSTM stack appears to maintain complementary information at both depths, and LAYA consistently treats the two layers as jointly relevant for classification.

Overall, the unified class-wise profiles confirm that LAYA adapts its use of depth to the structure of each domain: it develops distinct attention fingerprints per class in Fashion-MNIST, collapses onto the deepest representation in CIFAR-10, and adopts a balanced, two-layer strategy on IMDB.

Layer-wise attention in artwork classification. Figures 3 and 4 summarize the global and class-wise attention statistics of LAYA on the *Best Artworks of All Time* dataset. Taken together, these results indicate that LAYA does not treat depth in a purely monotonic, "deeper-is-better" fashion, but instead discovers a sparse, task-specific allocation of importance across ViT layers. The global profile in Fig. 3 shows that attention concentrates on a subset of intermediate and late layers (notably h_5, h_7, and h_9–h_{11}). In contrast, very shallow and very deep layers receive negligible mass. This pattern is consistent with the intuition that painterly style depends on mid-level cues such as texture, brushstroke statistics, and color composition, as well as higher-level structural cues, rather than on the most abstract semantics alone. The class-wise heatmaps in Fig. 4 further reveal distinctive "style fingerprints" in attention space: movements such as Byzantine Art and Proto Renaissance place substantial weight on earlier blocks, whereas Cubism, Neoplasticism, and Pop Art rely more heavily on later ones, reflecting their differing degrees of geometric abstraction and global structure. Correct predictions exhibit sharper, more characteristic attention patterns per style. At the same time, misclassified

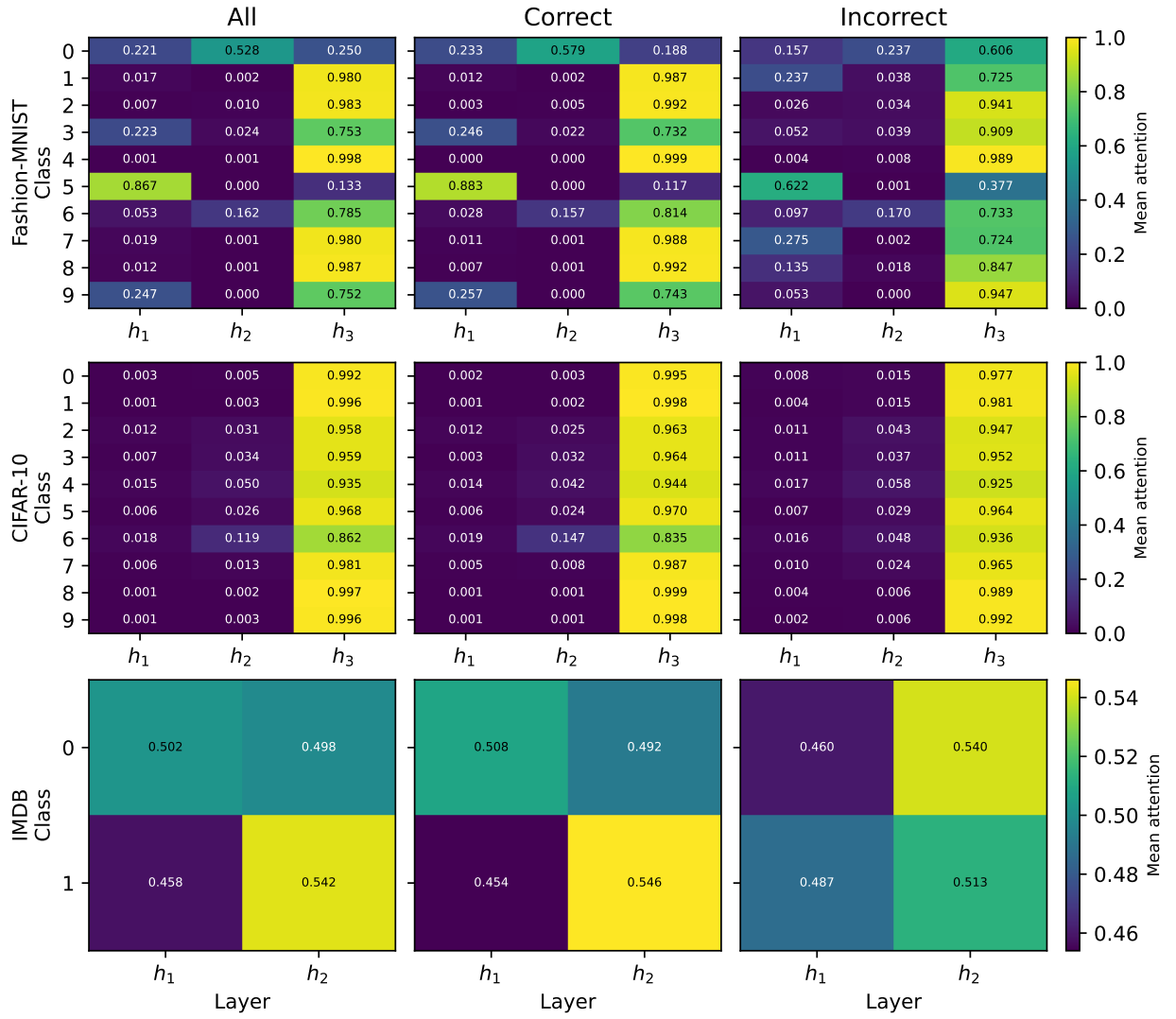

Figure 2: Class-wise mean attention profiles of LAYA across datasets. Rows correspond to Fashion-MNIST (top), CIFAR-10 (middle), and IMDB (bottom), while columns correspond to all test samples (left), correctly classified samples (center), and misclassified samples (right). Each cell reports the average attention $\bar{\alpha}_{c,i}$ assigned to layer i for class c.

samples tend to show more diffuse or shifted depth usage, suggesting that deviations from the canonical style-specific profile indicate uncertainty or confusion.

5 Conclusion and Future Work

This work revisited the conventional design choice of relying solely on the final hidden representation for prediction in deep networks. The proposed LAYA shows that the output stage can actively leverage the complete representational hierarchy by assigning input-dependent attention weights across depth. Experiments on both vision and language benchmarks indicate that this strategy leads to consistent improvements in predictive performance, while simultaneously providing intrinsic interpretability through layer-attribution scores that quantify the contribution of each abstraction level to the final decision.

Despite operating solely at the output stage, LAYA yields attention profiles that reveal clear and structured depth preferences. These profiles provide a principled basis for *attention-guided model compression*: layers that consistently

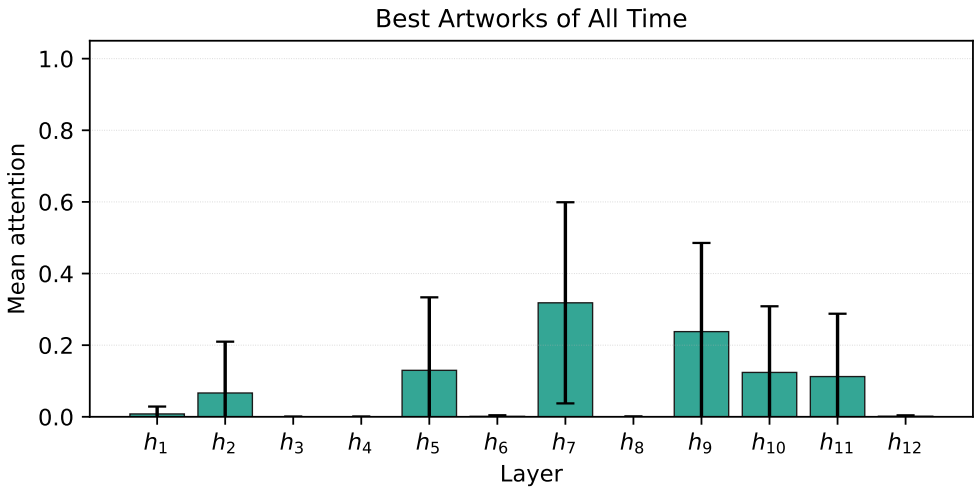

Figure 3: Layer-wise attention statistics for the *Best Artworks of All Time* dataset (mean $\pm$ standard deviation over the full test set). The attention distribution reveals heterogeneous depth contributions, with notable peaks around intermediate layers and sparse activations elsewhere.

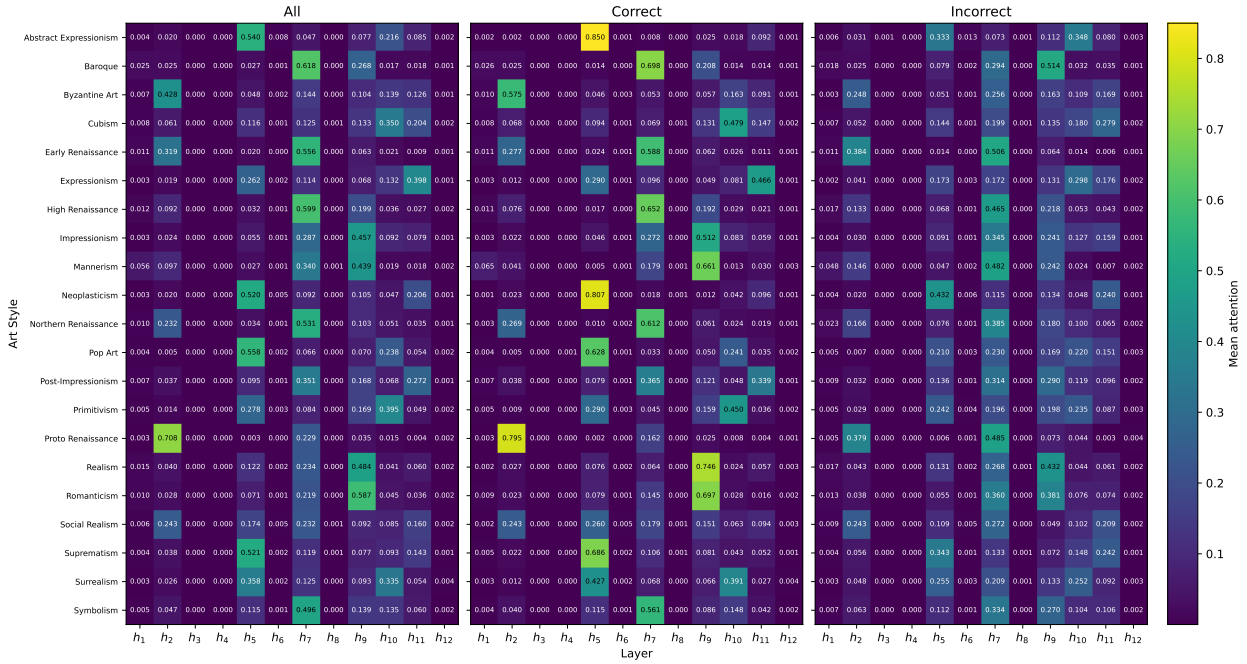

Figure 4: Class-wise mean attention profiles of LAYA on the *Best Artworks of All Time* dataset. Columns correspond to all test samples (left), correctly classified samples (center), and misclassified samples (right). Each cell reports the average attention $\bar{\alpha}_{c,i}$ assigned to layer i for class (style) c.

receive negligible attention may be redundant for the downstream task and thus candidates for simplification or removal, particularly when paired with fine-tuning or structural pruning.

Beyond compression, several directions emerge in which depth-aware output reasoning may have a broader impact. First, the observation that LAYA sometimes concentrates attention on intermediate representations provides a principled basis for *early-exit* strategies, in which inference is terminated once sufficient evidence has accumulated at earlier depths. Such adaptive computation can reduce inference costs while offering a transparent justification for the exit point. Second, systematic differences in attention profiles between correctly and misclassified samples suggest potential

diagnostic applications, such as identifying complex or out-of-distribution examples, highlighting problematic regions of the input space, or guiding dataset curation. Finally, the layer-attribution scores produced by LAYA offer new opportunities for *probing the role of depth* across architectures and tasks, providing insights into layer specialization and the degree to which different problems rely on shallow versus deep abstractions.

Overall, these findings suggest that treating the output stage as a depth-aware aggregation module—rather than a passive projection—offers a simple yet powerful way to leverage internal representations. LAYA provides a concrete demonstration of this principle and opens new opportunities in efficient inference, diagnostic interpretability, and depth-centric analysis of neural networks.

Acknowledgments

The author heartfully acknowledges Giovanna Castellano for valuable discussions that helped clarify and refine key aspects of this work, and for her constructive feedback on the manuscript.

References

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. UNet++: A nested U-Net architecture for medical image segmentation. In *International workshop on deep learning in medical image analysis*, pages 3–11. Springer, 2018.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

Ranran Huang, Yu Wang, and Huazhong Yang. Cross-layer attention network for fine-grained visual categorization. *arXiv preprint arXiv:2210.08784*, 2022.

Yafang Zheng, Lei Lin, Shuangtao Li, Yuxuan Yuan, Zhaohong Lai, Shan Liu, Biao Fu, Yidong Chen, and Xiaodong Shi. Layer-wise representation fusion for compositional generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19706–19714, 2024.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 2464–2469. IEEE, 2016.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. Pmlr, 2015.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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, 2016.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

D Hendrycks. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International conference on learning representations (ICLR)*, 2015.