
Activation with Intrinsic-Extrinsic Consensus

Tian Qiu^{1 2} Zunlei Feng³ Yang Gao^{1 2} Bingde Hu^{1 4 *} Yi Gao¹ Mingli Song^{1 2}

Abstract

Artificial Neural Networks (ANNs) are powerful tools for complex decision-making tasks. While existing activation mechanisms often promote sparsity through thresholding, they lack explicit awareness of feature channel relevance, causing networks to continuously suffer from interference by noisy channels. Such irrelevant activation signals can propagate through the network and adversely affect the final decision. Inspired by observations that channel relevance can be reflected in both intrinsic activity levels and extrinsic decision weights, and that there is strong consensus between these two aspects, we propose AIEC (Activation with Intrinsic-Extrinsic Consensus), a novel activation mechanism that has the ability to identify and suppress irrelevant feature channels during training. With a basic threshold activation, AIEC integrates an intrinsic Activation-Counting Unit that tracks channel activation statistics, an extrinsic Decision-Making Unit that learns channel decision weights, and a Consensus Gatekeeping Unit that suppresses irrelevant channels based on the agreement between intrinsic and extrinsic channel relevance assessments. Extensive experiments demonstrate that AIEC can effectively suppress irrelevant channels and encourage sparser representations. Furthermore, AIEC is compatible with a wide range of mainstream ANN architectures and achieves superior performance compared to existing activation mechanisms across multiple tasks and domains.

Code is available at <https://github.com/horrible-dong/AIEC>.

¹State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, Zhejiang, China ²Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou, Zhejiang, China ³School of Software Technology, Zhejiang University, Ningbo, Zhejiang, China ⁴Bangsun Technology Co., Ltd., Hangzhou, Zhejiang, China. *Correspondence to: Bingde Hu <tonyhu@zju.edu.cn>.

1. Introduction

Artificial Neural Networks (ANNs) (LeCun et al., 2015) have demonstrated remarkable capabilities in dealing with complex decision-making problems. From original Multi-Layer Perceptron (MLP) (McClelland et al., 1987) to classical Convolutional Neural Networks (CNNs) (LeCun et al., 1998) and trendsetting Vision Transformers (ViTs) (Han et al., 2022), their evolving architectures have driven breakthroughs in computer vision (Voulodimos et al., 2018) and beyond (Zhou et al., 2022; Ren et al., 2023).

As a core component of ANNs’ feature learning, activation mechanisms like ReLU (Glorot et al., 2011) employ thresholding to sparsify representations, achieving preliminary feature selection and offering advantages such as information disentanglement, linear separability, and potential generalization ability (Glorot et al., 2011). However, such mechanisms are static, relying solely on instantaneous input intensity and lacking explicit awareness of feature channel relevance: on one hand, they struggle to distinguish transient noise from genuine features; on the other hand, they remain unaware of channels’ actual contributions to the network’s decision. This limitation makes it difficult to effectively prevent noise interference from irrelevant channels. The erroneous activation of these channels may be amplified by subsequent layers, ultimately affecting the network’s final decision. This highlights the urgent need for an activation mechanism with enhanced context-awareness.

As observed in Figure 1, channel relevance can be measured from both intrinsic and extrinsic aspects. Intrinsically, under stimuli of homogeneous samples, some channels frequently activate and contribute to the network’s decision, whereas the majority of channels, though rarely active, should ideally remain entirely silent. Extrinsically, the decision weights of the linear classifier acting on the post-activation feature vector can also reflect each channel’s relevance (contribution) to the network’s final decision (prediction). Moreover, a strong consistency is observed between the intrinsic and extrinsic aspects (b1 and b2). This consensus can guide the network in assessing channel relevance and further suppressing interference from less relevant channels.

To address the lack of channel relevance assessment in existing activation mechanisms, and based on the above observations, we propose AIEC (Activation with Intrinsic-Extrinsic

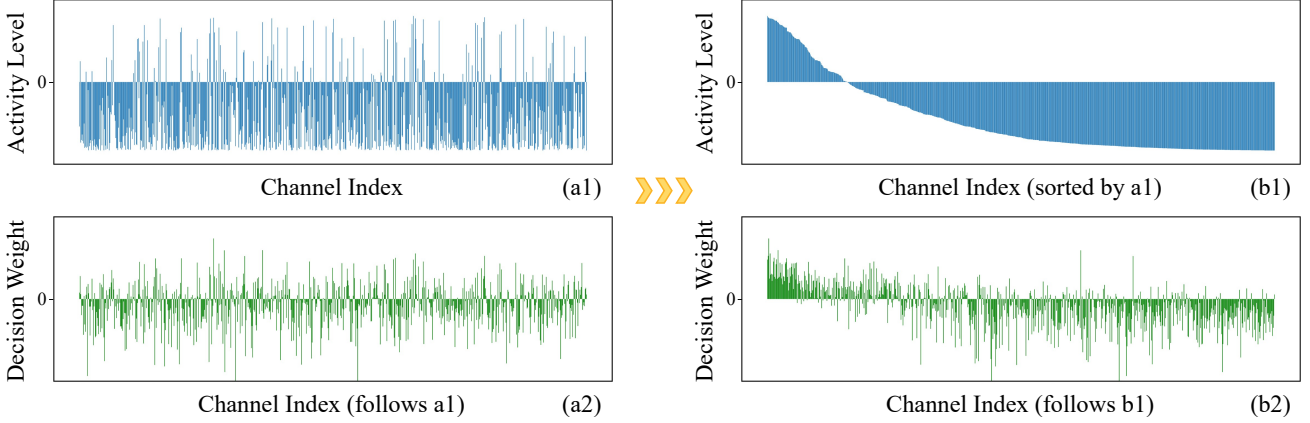


Figure 1. The intrinsic activity level (a1) and extrinsic decision weight (a2) are recorded for each channel of the feature vector after ReLU in the final block of ViT-Tiny, specifically for the “truck” category in the CIFAR-10 dataset. The activity level of a channel is computed based on its historical activation statistics over samples from the “truck” category, defined as “(number of activations - number of inhibitions) / total sample count”. The decision weight is learned as the weight of a linear classifier applied to the post-activation feature vector. (b1) is (a1) sorted in descending order of activity level, and (b2) is (a2) re-indexed following the index order of (b1).

Consensus), a novel activation mechanism with awareness of channel relevance that can identify and suppress irrelevant feature channels. AIEC integrates an online assessment established through intrinsic-extrinsic consensus to identify irrelevant channels. At the intrinsic aspect, AIEC tracks threshold-activation statistics for each channel across different categories in real time, establishing a channel relevance assessment based on channels’ activity levels. At the extrinsic aspect, AIEC learns decision weights for each channel through supervised feedback, constructing a channel relevance assessment based on channels’ influence on the network’s final decision. The final criterion for irrelevant channel identification is the consensus between the intrinsic and extrinsic assessments. Guided by this consensus, AIEC performs channel-wise gatekeeping for noise cleaning through a regularization loss, effectively suppressing activation responses from irrelevant channels while preserving those from relevant channels. Extensive experiments demonstrate the outstanding performance of AIEC compared to existing activation mechanisms across various mainstream ANN architectures, datasets, tasks and domains.

The contributions of this paper are summarized as follows:

- The observations that channel relevance can be assessed from two dimensions: intrinsic activity levels and extrinsic decision weights, and that a high consensus exists between them. This consensus provides a reliable basis for assessing channel relevance, an aspect overlooked by existing activation mechanisms.
- The proposal of AIEC (Activation with Intrinsic-Extrinsic Consensus), a novel activation mechanism that identifies irrelevant feature channels based on the consensus of intrinsic and extrinsic aspects, and performs channel-wise gatekeeping for targeted noise cleaning.

- Extensive experiments demonstrating the superior performance of AIEC over existing activation mechanisms across diverse mainstream ANN architectures, datasets, as well as multiple tasks and domains.

2. Observations

This section presents observations on ANN’s channel activation from both intrinsic and extrinsic aspects, providing insights for improving existing activation mechanisms.

Observation 1: *Variance in channel activity levels intrinsically reflects channel relevance.*

Under threshold-based activation (e.g., ReLU), relevant signals get activated while irrelevant ones are inhibited. As illustrated in Figure 1(a1), when numerous homogeneous samples are input, different channels show different activity levels, which is deemed to be positively correlated with the channels’ relevance. Also in Figure 1(a1), for classification tasks, a category is associated with only a sparse and specific set of channels, indicating the presence of a significant proportion of redundant/irrelevant channels in the network, and ideally, these irrelevant channels should remain silent.

Observation 2: *Linear classifier acting on activated features learns distinct decision weights for each channel, extrinsically indicating channel relevance.*

For classification tasks, we introduce a linear classifier (i.e., learnable linear decision weights $\{\mathbf{w}^k \in \mathbb{R}^C\}_{k=1}^K$ without the bias term, where K is the number of classes and C is the number of channels) on the post-activation feature vector $\mathbf{a}^k \in \mathbb{R}_{[0,+\infty]}^C$ with C channels belonging to the k -th class. The learned decision weights (i.e., the linear weights $\mathbf{w}^k \in \mathbb{R}^C$) per channel are shown in Figure 1(a2). It is

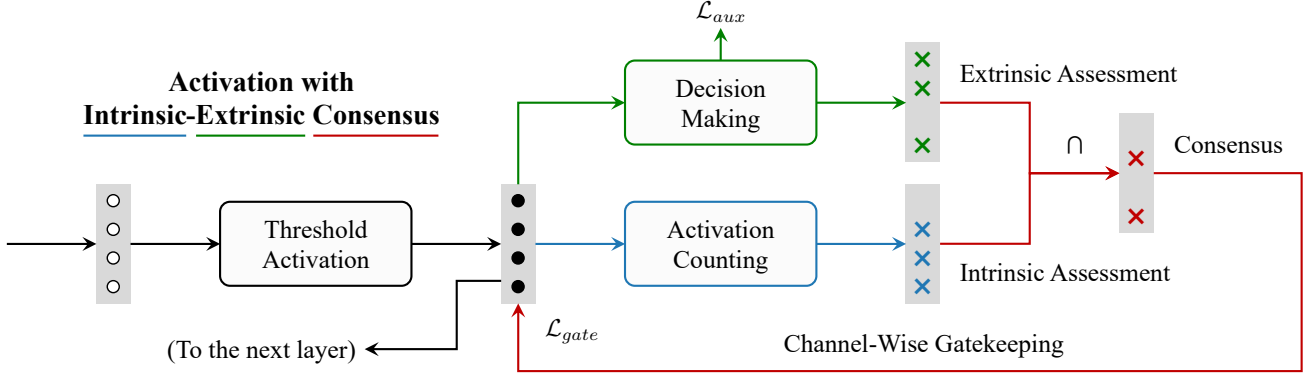


Figure 2. The proposed AIEC (Activation with Intrinsic-Extrinsic Consensus) employs a basic threshold activation that provides a “prior” partition boundary (activation threshold) to divide the channels of a feature vector into irrelevant and relevant groups. Building upon this, intrinsic and extrinsic channel relevance assessments are derived respectively through activation-counting and decision-making processes in a data-driven manner. Finally, a “posterior” partition boundary between irrelevant and relevant channels is established based on intrinsic-extrinsic consensus, enabling channel-wise gatekeeping that suppresses irrelevant channels through a regularization loss.

observed that only a small subset of channels acquires high weights. The underlying mechanism is that: the predicted score (i.e., the logit $\mathbf{a}^k \cdot \mathbf{w}^{kT} \in \mathbb{R}$) is the weighted summation of the non-negative activation values of all channels and their corresponding weights. To increase the predicted score for the target class, the linear classifier tends to assign higher weights to important channels to amplify their impact. Consequently, the weights learned by the linear classifier can quantify the relevance of channels to the network’s final decision from an extrinsic view.

Observation 3: *Intrinsic and extrinsic aspects show consensus for channel relevance assessment.*

In Figure 1, we sort (a1) in descending order of activity level to obtain (b1), and then re-index (a2) following the index order of (b1) to obtain (b2). It can be observed that the intrinsic and extrinsic views (b1, b2) exhibit consistency, suggesting that their consensus may serve as a reliable basis for channel relevance assessment. Quantitative correlation analysis between (b1) and (b2) is shown in Appendix §A.1.

3. Methodology

As illustrated in Figure 2, we propose AIEC (Activation with Intrinsic-Extrinsic Consensus), a novel activation mechanism with channel relevance awareness. AIEC consists of four components: (1) a basic Threshold Activation Unit (TAU), (2) an intrinsic Activation-Counting Unit (ACU) that tracks each channel’s activation statistics and provides intrinsic channel relevance assessment, (3) an extrinsic Decision-Making Unit (DMU) that learns each channel’s decision weight and provides extrinsic channel relevance assessment, and (4) a Consensus Gatekeeping Unit (CGU) that pinpoints irrelevant channels based on the consensus of intrinsic ACU and extrinsic DMU, and then suppresses these irrelevant channels through a regularization loss.

3.1. Threshold Activation Unit

As a basic part, Threshold Activation Unit (TAU) functions as a gating mechanism for signal transmission. Given a non-negative activation threshold τ , the input signal x is allowed to pass to the next layer as it exceeds the threshold τ , otherwise it gets inhibited to zero. The TAU’s operation on input x can be expressed as

$$\text{TAU}(x) = \begin{cases} x, & \text{if } x > \tau \\ 0, & \text{if } x \leq \tau \end{cases}. \quad (1)$$

Consider a pre-activation feature map $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$ with C channels. The post-activation feature map $\mathbf{A} \in \mathbb{R}_{[0,+\infty]}^{H \times W \times C}$ is obtained by applying the TAU element-wise to $\mathbf{F}$, whose elements are all non-negative:

$$\mathbf{A} = \text{TAU}(\mathbf{F}). \quad (2)$$

While TAU brings some sparsity to feature representation, which can be viewed as a form of “prior” partitioning, its effectiveness in identifying truly task-irrelevant channels and inhibiting their noise remains limited. Subsequent sections show how to overcome this by leveraging the consensus of intrinsic and extrinsic channel relevance assessment for sharper “posterior” suppression.

3.2. Intrinsic and Extrinsic Assessments

Operating on the post-activation feature at the channel level, this section aims to identify irrelevant channels from the intrinsic and extrinsic views, respectively.

First, perform global average pooling on the post-activation feature map $\mathbf{A}$ to obtain the global feature vector $\mathbf{a} \in \mathbb{R}_{[0,+\infty]}^C$, whose elements are all non-negative:

$$\mathbf{a} = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W \mathbf{A}_{h,w}. \quad (3)$$

For intrinsic channel relevance assessment, Activation-Counting Unit (ACU) intrinsically quantifies channel relevance by statistically measuring each channel's activity level. For a given dataset with K categories, an ACU maintains three counters for each category: $\{\theta^k, \eta^k, \phi^k\}_{k=1}^K$. Over samples from category k , $\theta^k \in \mathbb{N}^C$, $\eta^k \in \mathbb{N}^C$, and $\phi^k \in \mathbb{N}^C$ tracks each channel's historical activation statistics: the number of activations after TAU, the number of inhibitions after TAU, and the total sample count. Specifically, for an input data of category k , denote $\mathbf{a}^k \in \mathbb{R}_{[0,+\infty]}^C$ as its channel-level activated feature (Eq. 3). Since we only focus on whether a channel is activated or not, $\mathbf{a}^k$ is binarized to $\tilde{\mathbf{a}}^k \in \mathbb{I}^C$ as follows:

$$\tilde{\mathbf{a}}_c^k = \begin{cases} 1, & \text{if } \mathbf{a}_c^k > 0 \\ 0, & \text{if } \mathbf{a}_c^k \leq 0 \end{cases}. \quad (4)$$

Next, based on $\tilde{\mathbf{a}}^k$, the ACU updates the activation statistics θ^k , η^k , and ϕ^k as follows:

$$\theta^{k,t} = \theta^{k,t-1} + \tilde{\mathbf{a}}^k, \quad (5)$$

$$\eta^{k,t} = \eta^{k,t-1} + (1 - \tilde{\mathbf{a}}^k), \quad (6)$$

$$\phi^{k,t} = \phi^{k,t-1} + \mathbf{1}, \quad (7)$$

where $t-1$ and t denote previous and current iteration, respectively. $\mathbf{1} \in \mathbb{I}^C$ is a vector full of one. In implementation, only samples enabling correct predictions from the Decision-Making Unit described below are included in the statistics to obtain more accurate results. From these statistics, we define "relative firing rate" $\mathbf{v}^k \in \mathbb{R}_{[-1,1]}^C$, which quantifies each channel's activity level for category k :

$$\mathbf{v}^{k,t} = \frac{\theta^{k,t} - \eta^{k,t}}{\phi^{k,t}}. \quad (8)$$

According to the principle of threshold activation, signals from irrelevant channels tend to be inhibited when passing through the TAU, resulting in these channels' low activity level. The intrinsic channel relevance division for each category is represented by $\{\mathcal{G}_c^{k,Intr}\}_{c=1}^K$, where *Intr* denotes the term "Intrinsic". With zero as the natural demarcation point, $\mathcal{G}_c^{k,Intr} \in \mathbb{I}^C$ can be established as

$$\mathcal{G}_c^{k,Intr} = \begin{cases} 0, & \text{if } \mathbf{v}_c^k > 0 \\ 1, & \text{if } \mathbf{v}_c^k \leq 0 \end{cases}, \quad (9)$$

where the indicator "1" signifies potential **irrelevant** channels.

For extrinsic channel relevance assessment, Decision-Making Unit (DMU) extrinsically quantifies channel relevance from the natural reflection of the network's decision. A DMU contains K learnable linear decision weights $\{\mathbf{w}^k\}_{k=1}^K$, respectively for the K categories, where $\mathbf{w}^k \in$

$\mathbb{R}^C$. For an input data of category k , denote $\mathbf{a}^k \in \mathbb{R}_{[0,+\infty]}^C$ as its channel-level activated feature (Eq. 3). Then, linear projection and softmax are applied to $\mathbf{a}^k$ to obtain the intermediate predictions $\{p^k\}_{k=1}^K$ for each category:

$$\{p^k\}_{k=1}^K = \text{softmax}(\mathbf{a}^k \cdot \{\mathbf{w}^{k,T}\}_{k=1}^K), \quad (10)$$

where $p^k \in [0, 1]$ is the predicted probability for category k . After that, compute the auxiliary loss aiming to minimize the difference between the prediction p^k and its label $y^k \in \{0, 1\}$ as follows:

$$\mathcal{L}_{aux} = -\sum_{k=1}^K y^k \log(p^k). \quad (11)$$

The decision weight $\mathbf{w}^k \in \mathbb{R}^C$ being learned through $\mathcal{L}_{aux}$ continuously quantifies each channel's contribution/relevance to the network's decision for category k . The extrinsic channel relevance division for each category is represented by $\{\mathcal{G}_c^{k,Extr}\}_{c=1}^K$, where *Extr* denotes the term "Extrinsic". With zero as the natural demarcation point, $\mathcal{G}_c^{k,Extr} \in \mathbb{I}^C$ can be established as

$$\mathcal{G}_c^{k,Extr} = \begin{cases} 0, & \text{if } \mathbf{w}_c^k > 0 \\ 1, & \text{if } \mathbf{w}_c^k \leq 0 \end{cases}, \quad (12)$$

where the indicator "1" signifies potential **irrelevant** channels.

3.3. Gatekeeping with Consensus

With the consensus of intrinsic and extrinsic channel relevance assessment, Consensus Gatekeeping Unit (CGU) is performed to clean the responses from irrelevant channels during training. The final channel relevance division $\{\mathcal{G}_c^k\}_{c=1}^K$ for each category is the intersection (i.e., logical "and") of the intrinsic division and the extrinsic division:

$$\{\mathcal{G}_c^k\}_{c=1}^K = \{\mathcal{G}_c^{k,Intr} \cap \mathcal{G}_c^{k,Extr}\}_{c=1}^K. \quad (13)$$

Based on this consensus, given an input data of category k , first filter out **irrelevant** channels from $\mathbf{a}^k$ by $\mathcal{G}^k$:

$$\tilde{\mathbf{a}}^k = \mathbf{a}^k \odot \mathcal{G}^k, \quad (14)$$

where $\odot$ denotes the Hadamard Product. Then, impose gatekeeping on the filtered **irrelevant** channels by constructing a new regularization loss item $\mathcal{L}_{gate}$ as follows:

$$\mathcal{L}_{gate} = \frac{\sum_{c=1}^C \|\tilde{\mathbf{a}}_c^k\|_2}{\sum_{c=1}^C \mathcal{G}_c^k}. \quad (15)$$

In this way, only the activation responses from channels with low relevance will be gradually suppressed, while the ones from other channels with high relevance are preserved.

3.4. Neural Network Learning

The proposed AIEC (Activation with Intrinsic-Extrinsic Consensus) can replace the network’s original activation mechanisms. The gatekeeping applies to the network’s activated global feature. Networks that extract feature maps/sequences compute the global feature by taking the global average of the feature maps/sequences along the channels. Regarding some Transformer models that incorporate a class token, simply peel off the class token separately as the global feature while applying the corresponding gatekeeping. Additionally, for Transformer models, the proposed AIEC is applied to each block, as Transformers excel in capturing global context throughout, while for CNN models, AIEC is applied to the last block since high-level semantics only exist in deep representations (Raghu et al., 2021). The final loss $\mathcal{L}$ is expressed as

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_{aux} \cdot \frac{1}{N} \sum_{n=1}^N \mathcal{L}_{aux}^n + \lambda_{gate} \cdot \frac{1}{N} \sum_{n=1}^N \mathcal{L}_{gate}^n, \quad (16)$$

where $\mathcal{L}_{task}$ is the primary loss for the specific task; for example, in classification tasks, $\mathcal{L}_{task}$ represents the cross-entropy loss. N is the number of layers in the network that have AIEC applied, and $\lambda_{\{aux, gate\}}$ are balancing parameters. Note that ACU, DMU, and CGU only work during the training phase, and in the inference phase, only TAU needs to be involved. Descriptions of the training and inference algorithms can be found in Appendix §A.4.

4. Experimental Study

Datasets. We adopt seven datasets, including four vision datasets: CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), ImageNet-100 (Deng et al., 2009), and ImageNet-1K (Deng et al., 2009), and three non-vision datasets: Elliptic (Weber et al., 2019), T-Finance (Tang et al., 2022), and Weibo21 (Nan et al., 2021), to verify the effectiveness of the proposed AIEC.

Compared methods. The proposed AIEC is compared with different types of mainstream activation mechanisms mentioned in Related Work §5.1, including Softplus (Dugas et al., 2000), ELU (Clevert et al., 2015), SELU (Klambauer et al., 2017), SiLU (Ramachandran et al., 2017), ReLU (Glorot et al., 2011), GELU (Hendrycks & Gimpel, 2016), and GDN (Ballé et al., 2015).

Experimental settings. The image size of CIFAR-{10,100} remains 32×32, while the images in ImageNet-{100,1K} are uniformly scaled to 224×224. To ensure the generality of the network, the activation threshold τ is uniformly set to 0. The balancing parameter λ_{aux} for the auxiliary loss $\mathcal{L}_{aux}$ is empirically set to 1 to match the magnitude of the task loss $\mathcal{L}_{task}$, and the balancing parameter λ_{gate} for the gatekeeping loss $\mathcal{L}_{gate}$ varies according to different network architectures and datasets as discussed in Appendix

§A.2. All experiments use the same data augmentations provided by timm (Wightman, 2019), AdamW optimizer with weight decay of 0.05, drop-path rate of 0.1, gradient clipping norm of 1.0, and cosine annealing learning rate scheduler with linear warm-up. All experiments are trained for 300 epochs from scratch. The automatic mixed precision training strategy is adopted to speed up the training. All other training settings, including batch size, learning rate, warm-up epochs, and so on, are kept identical throughout each set of comparative experiments. Note that the numerical results are the average under three different random seeds, and no pre-training is used.

4.1. AIEC on ViTs and CNNs

The proposed AIEC can be incorporated into popular Vision Transformer (ViT) and its variants. Table 1 shows the top-1 accuracy (%) across CIFAR-{10,100} and ImageNet-{100,1K} using the proposed AIEC on five different ViT architectures: ViT (Dosovitskiy et al., 2020), DeiT (Touvron et al., 2021a), CaiT (Touvron et al., 2021b), PVT (Wang et al., 2021), and TNT (Han et al., 2021). The proposed AIEC can replace all the existing activations in each block. The results consistently illustrate that AIEC outperforms the baselines, showcasing its robustness.

The proposed AIEC is also evaluated on various mainstream CNNs, including AlexNet (Krizhevsky et al., 2017), VGG (Simonyan & Zisserman, 2014), MobileNet (Howard et al., 2017), ShuffleNet(V2) (Ma et al., 2018), and ResNet (He et al., 2016). The proposed AIEC replaces the original activations in the last block since previous works have shown that high-level semantics in CNNs only exist in deep representations (Raghu et al., 2021). The results in Table 2 highlight the versatility and robustness of AIEC in handling diverse CNN architectures and datasets.

4.2. Generalization to Other Tasks

The proposed AIEC can also perform various other tasks or domains, like the non-vision task anomaly detection, including finance fraud detection and fake news detection. For finance fraud detection, the Elliptic (Weber et al., 2019) and T-Finance (Tang et al., 2022) datasets are employed. For fake news detection, the Weibo21 (Nan et al., 2021) dataset is employed. GCN (Kipf & Welling, 2016) and GraphSAGE (GSAGE) (Hamilton et al., 2017) are chosen networks. The results are shown in Table 3.

4.3. Computational Costs

As mentioned before, ACU, DMU, and CGU are active only during training, while in inference, only TAU is required. Table 4 presents the computational costs during training and inference regarding “GPU Memory (GiB)” and “Latency

Table 1. Top-1 accuracy (%) across the CIFAR-10, CIFAR-100, ImageNet-100, and ImageNet-1K datasets using the proposed AIEC on Vision Transformer (ViT) and its variants.

Top-1 Acc / %		ViT-Tiny	DeiT-Tiny	CaiT-XXS	PVT-Tiny	TNT-Small
CIFAR-10	Softplus	84.3	84.7	82.5	90.6	88.3
	ELU	82.0	81.4	80.7	89.3	85.4
	SELU	79.4	79.9	78.4	85.4	83.7
	SiLU	85.5	86.6	86.6	92.5	90.5
	ReLU	89.9	89.6	89.4	93.0	90.8
	GELU	89.2	89.2	88.7	92.8	91.1
	GDN	81.8	83.0	80.0	82.8	85.1
	AIEC	91.8	91.9	90.5	94.0	92.7
CIFAR-100	Softplus	62.4	63.4	60.4	69.5	65.2
	ELU	60.0	60.0	59.3	69.3	63.8
	SELU	57.5	58.3	55.8	65.7	60.9
	SiLU	65.5	67.1	63.9	70.2	65.1
	ReLU	65.7	67.0	65.8	70.9	65.4
	GELU	65.4	67.0	65.5	70.6	64.4
	GDN	59.4	59.8	56.2	64.4	62.5
	AIEC	71.1	71.8	70.0	76.0	73.9
ImageNet-100	Softplus	74.1	75.3	70.9	79.5	78.9
	ELU	68.9	69.4	69.1	77.1	79.3
	SELU	66.4	67.0	65.9	76.1	76.4
	SiLU	74.1	75.1	76.1	79.5	77.6
	ReLU	75.4	75.6	76.0	81.9	79.9
	GELU	76.4	74.6	76.7	81.4	77.2
	GDN	67.9	66.3	69.5	75.8	76.9
	AIEC	82.1	82.5	81.3	86.3	86.5
ImageNet-1K	Softplus	70.0	71.9	70.3	71.5	72.0
	ELU	64.2	67.9	68.1	69.2	70.7
	SELU	63.1	66.2	66.7	68.5	70.3
	SiLU	66.9	72.0	73.2	72.8	71.5
	ReLU	70.9	73.2	74.0	73.7	73.4
	GELU	70.4	73.0	73.6	73.5	73.3
	GDN	65.2	66.4	66.1	66.5	68.2
	AIEC	73.0	73.7	74.1	74.6	78.1

Table 3. Generalization performance of the proposed AIEC on the non-vision anomaly detection tasks including finance fraud detection and fake news detection.

Dataset	T-Finance			Elliptic			Weibo		
	AUC	AP	R@K	AUC	AP	R@K	AUC	AP	R@K
GCN-ReLU	92.9	75.8	70.6	81.1	21.3	25.4	98.5	94.0	90.2
GCN-GELU	93.0	76.2	72.8	77.4	15.2	10.1	98.6	94.0	89.9
GCN-AIEC	93.1	77.8	73.4	80.9	34.9	42.2	98.6	95.1	90.2
GSAGE-ReLU	84.5	60.3	63.7	84.9	35.0	38.2	96.6	92.2	88.2
GSAGE-GELU	84.9	52.9	58.5	85.0	35.1	41.0	97.3	93.3	89.3
GSAGE-AIEC	90.9	70.0	67.3	85.6	43.1	46.1	98.3	93.4	89.3

(s)” (the average time it takes for a network to process a batch of data). Notably, the activation methods used in original networks should be implemented manually as our AIEC does. Using the methods directly from pre-made libraries (like torch.nn) can result in unfair comparisons due to their high optimization at the low level. Table 4 shows that AIEC’s training GPU overhead is negligible and its inference speed is on par with other methods. Further analysis of computational complexity is provided in Appendix §A.3.

4.4. Visualization Results

As shown in Figure 3, with the proposed AIEC, the activation responses become sparser, as can be seen from the activity levels of the channels, implying that some irrelevant

Table 2. Top-1 accuracy (%) across the CIFAR-10, CIFAR-100, ImageNet-100, and ImageNet-1K datasets using the proposed AIEC on various CNN architectures.

Top-1 Acc / %		AlexNet	VGG-11	MobileNet	ShuffleNet	ResNet-18
CIFAR-10	Softplus	85.6	91.3	87.4	89.2	94.6
	ELU	86.1	92.0	87.7	89.0	94.7
	SELU	85.7	91.5	87.2	88.9	94.6
	SiLU	86.0	91.9	87.8	89.3	95.1
	ReLU	86.0	92.2	87.4	89.4	95.0
	GELU	85.8	91.9	87.4	89.3	94.9
	GDN	85.4	91.1	87.0	88.5	94.0
	AIEC	86.3	92.2	89.0	89.9	95.1
CIFAR-100	Softplus	57.6	69.6	65.4	66.2	75.5
	ELU	58.4	69.9	65.5	66.1	75.7
	SELU	58.1	69.7	65.6	65.9	75.6
	SiLU	58.1	69.9	65.2	66.3	76.1
	ReLU	57.2	70.2	66.0	66.3	75.7
	GELU	57.4	70.0	65.4	66.2	75.6
	GDN	56.8	70.1	64.8	65.6	74.3
	AIEC	58.9	71.2	66.1	66.9	77.0
ImageNet-100	Softplus	75.7	87.0	80.6	80.9	84.6
	ELU	76.0	87.3	79.3	80.9	84.4
	SELU	75.7	87.6	79.2	80.4	84.1
	SiLU	76.6	87.8	80.1	81.7	84.9
	ReLU	76.3	87.7	80.6	81.6	84.9
	GELU	76.3	87.5	80.5	81.6	84.7
	GDN	75.5	86.7	78.7	80.0	83.5
	AIEC	79.2	88.1	80.8	82.2	86.5
ImageNet-1K	Softplus	56.1	68.4	67.2	68.5	69.3
	ELU	56.3	68.2	66.7	68.3	69.4
	SELU	56.1	67.8	67.1	68.4	68.9
	SiLU	56.4	69.0	67.4	69.1	69.7
	ReLU	56.5	69.0	68.1	69.0	69.7
	GELU	56.4	69.1	68.2	68.9	69.4
	GDN	55.6	68.1	66.3	68.0	68.3
	AIEC	57.8	70.1	68.6	69.6	70.4

Table 4. Computational costs during training and inference regarding “GPU Memory (GiB)” and “Latency (s)”. The networks were fed 224×224-pixel images with a batch size of 1024 on an NVIDIA A6000 GPU. “Latency” refers to the average time it takes for a network to process a batch of data.

Computational Costs	ViT-Tiny			ResNet-18		
	ReLU	GELU	AIEC	ReLU	GELU	AIEC
Training GPU Mem. / GiB	32.2	28.3	32.5	24.3	27.9	24.8
Inference GPU Mem. / GiB	4.8	4.8	4.9	7.4	7.4	7.4
Training Latency / ms	540.7	640.9	592.0	417.5	437.6	421.9
Inference Latency / ms	9.2	10.2	9.2	7.5	8.1	7.5

channels are identified and suppressed. Moreover, the channel gatekeeping in AIEC leads to clearer sign boundaries in the decision weights, reflecting that the network becomes more confident in assigning importance to channels that consistently contribute to the correct decision. These phenomena suggest that AIEC improves the precision of key feature extraction, which reduces learning difficulty and increases interpretability. More visualizations and discussions are provided in Appendix §A.7.

4.5. Ablation Study

The ablation study conducted mainly focuses on the effects of $\mathcal{L}_{aux}$ and different combinations of the intrinsic and extrinsic assessments, as presented in Table 5. First,

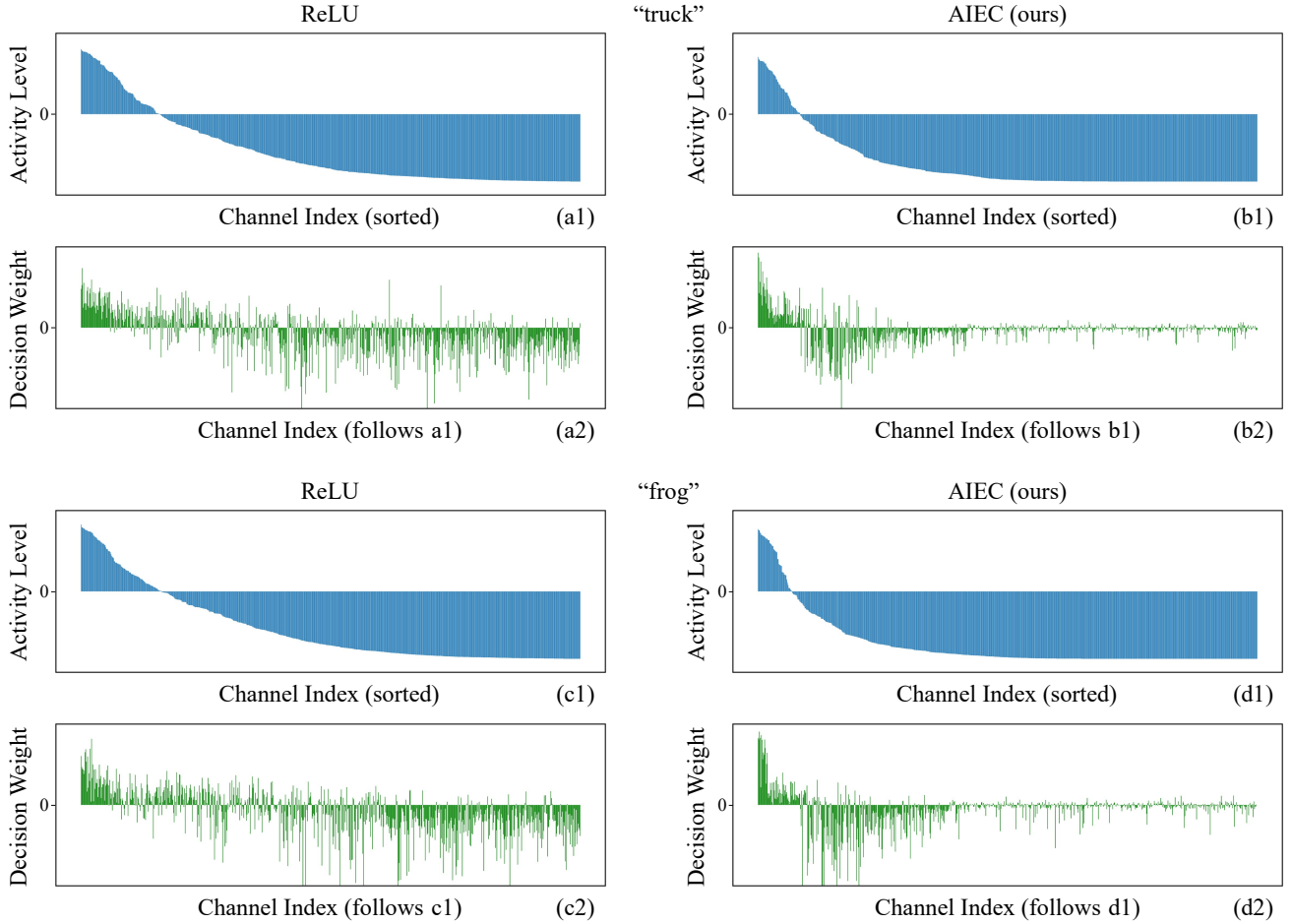


Figure 3. Visualization results. The intrinsic activity level (a1, b1) / (c1, d1) and extrinsic decision weight (a2, b2) / (c2, d2) are recorded for each channel of the feature vector after the activation (ReLU vs AIEC) in the last block of ViT-Tiny, specifically for the “truck” and “frog” categories in the CIFAR-10 dataset. More visualization results are provided in Figure 6, 7, 8, 9.

solely introducing $\mathcal{L}_{aux}$ has no effect, indicating that the actual performance contribution of AIEC comes from $\mathcal{L}_{gate}$. Besides, the performance with the intrinsic-extrinsic consensus (AIEC_{I∩E}) is better than using a single AIEC_I or AIEC_E. Moreover, The consensus of the two assessments (AIEC_{I∩E}) yields better results than the combination of both (AIEC_{I∪E}), as the consensus can reduce misjudgments. Finally, indiscriminately suppressing all channels (AIEC_{all}) yields no performance improvement.

4.6. Comparison with Other Methods

Mainstream activation mechanisms currently lack awareness of feature relevance. Against this backdrop, the design of AIEC therefore serves as a new benchmark, representing a novel form of activation that addresses this shortcoming. Apart from this, we also discuss and compare the proposed AIEC with other related baselines including:

L_0 regularization. Theoretically, the L_0 regularization technique (Louizos et al., 2017) can constrain the number of

Table 5. Ablation study of the proposed AIEC on CIFAR-100. AIEC _{$\mathcal{L}_{aux}$} denotes using the auxiliary loss $\mathcal{L}_{aux}$ only (without the gatekeeping loss $\mathcal{L}_{gate}$). AIEC_I, AIEC_E, AIEC_{I∪E}, and AIEC_{I∩E} denote AIEC with the channel relevance assessment in the form of intrinsic only, extrinsic only, intrinsic-extrinsic combination, and intrinsic-extrinsic consensus, respectively. AIEC_{all} denotes suppressing all channels indiscriminately.

Top-1 Acc / %	AIEC _{$\mathcal{L}_{aux}$}	AIEC _{all}	AIEC _I	AIEC _E	AIEC _{I∪E}	AIEC _{I∩E}
ViT-Tiny	66.7	65.8	69.7	68.5	69.6	71.1
ResNet-18	76.1	75.7	76.3	76.6	76.6	77.0

non-zero elements in features by adding a loss term. However, when the number of features is large, such a constraint makes it difficult for the model to determine which specific elements should be suppressed to zero. In contrast, the proposed AIEC provides clear criteria indicating which elements should be suppressed, offering more sufficient supervision signals and stronger interpretability.

Attention-based gating modules. Modules like SENet (Hu et al., 2018) and CBAM (Woo et al., 2018) introduce learn-

Table 6. Comparison with other methods on CIFAR-100.

Top-1 Acc / %	ReLU + L_0	GELU + L_0	SENet	CBAM	AIEC
ViT-Tiny	67.1	67.2	65.7	N/A	71.1
ResNet-18	76.5	76.2	75.6	76.2	77.0

able fully connected or convolutional layers to generate attention weights at the channel level (or the spatial level), but they simply weight the channels without evidence of sparse representation or interpretability. In contrast, the proposed AIEC leverages both intrinsic and extrinsic characteristics of threshold activation as direct supervision signals to identify irrelevant channels and apply noise cleaning targeted on them, thus providing strong sparsity and interpretability.

Table 6 presents the comparison of these methods. Note that CBAM is inapplicable to ViT-Tiny due to its convolutional layers. The results show that the proposed AIEC achieves superior performance compared to these methods.

5. Related Work

5.1. Activation Mechanisms

The activation mechanism (Nwankpa et al., 2018; Dubey et al., 2022) plays a pivotal role in ANNs as it defines how neurons respond to input signals and convert them into output signals. Activation mechanisms are categorized into different types, including logistic Sigmoid and Tanh variants (LeCun et al., 1998), Rectified Linear Unit variants (Glorot et al., 2011), Exponential Linear Unit variants (Clevert et al., 2015), Softplus variants (Dugas et al., 2000), probabilistic variants (Hendrycks & Gimpel, 2016), and others (Ballé et al., 2015; Ramachandran et al., 2017).

In the widely used form of activation (LeCun et al., 1998; Dugas et al., 2000; Glorot et al., 2011; Clevert et al., 2015; Hendrycks & Gimpel, 2016), features with low values are inhibited, and features with high values gain amplified influence according to the response rule of the neuron. Furthermore, some activation mechanisms (LeCun et al., 1998; Dugas et al., 2000; Glorot et al., 2011) can effectively achieve data sparsity, diminish redundant information, and enable better feature distinction. Additionally, the activation mechanisms, such as ELU (Clevert et al., 2015) and SiLU (Ramachandran et al., 2017) mentioned above, and others (Liu et al., 2020; Qiu et al., 2024b), contain learnable parameters. Parameters in Liu et al. (2020) can adapt to various data distributions, avoiding gradient vanishing and explosion, thereby enhancing the convergence speed and precision of ANNs. However, these extra parameters can only uniformly influence the response strength for all inputs, and are learned without explicit supervision. Consequently, like ordinary activation mechanisms, they still rely on instantaneous input intensity to gate feature propagation, lacking interpretable evidence for channel relevance awareness.

5.2. Feature Attribution Techniques

Feature attribution (Mandler & Weigand, 2024) aims to assess each feature’s contribution to the network’s final decision, distinguishing relevant from irrelevant features. Class Activation Mapping (CAM) (Zhou et al., 2016) computes the saliency map by projecting back the weights of the output layer onto the input feature maps. Grad-CAM (Selvaraju et al., 2017) extends CAM to any layer via the gradient relative to the target class. Built on this, Model Doctor (Feng et al., 2022) and DPA (Qiu et al., 2024a) respectively apply constraints on erroneous channel gradients and activated responses to correct the decision stream. Similarly, CAS (Bai et al., 2021) derives channel importance through the weights of a surrogate network and aligns the channel activation of perturbed samples with that of clean samples for improved adversarial robustness, while CIFS (Yan et al., 2021) extends this to a gradient-based version. GradToken (Cheng et al., 2025) exploits class-aware gradients to decouple the semantics and leverages class-spatial token relations to generate relevance maps. Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) decomposes decisions by propagating attributions backward through the network. Chefer et al. (2021), Wu & Ong (2021), and Ali et al. (2022) further extend and apply LRP in various tasks. Perturbation-based methods (Ivanovs et al., 2021) operate by manipulating input pixels and observing output changes. Geva et al. (2022), Deiseroth et al. (2023), and Vilas et al. (2023) project internal representations into a human-understandable class embedding space to uncover how predictions are constructed. FovEx (Panda et al., 2025) combines foveation-based transformations with gradient-driven overt attention to iteratively assess the relevance of each location.

These techniques inspire us to design an innovative activation mechanism with awareness of channel relevance based on each channel’s contribution to the network’s decision (prediction), and is able to effectively suppress interference from irrelevant channels.

6. Conclusion

In this paper, we propose AIEC (Activation with Intrinsic-Extrinsic Consensus), a novel activation mechanism with awareness of channel relevance that identifies and suppresses irrelevant feature channels during training based on the consensus between intrinsic activity levels and extrinsic decision weights. Through extensive experiments across vision and non-vision tasks, we demonstrate that AIEC consistently outperforms existing activation mechanisms, promotes sparser and more interpretable representations, and generalizes effectively across diverse architectures. Our work highlights the importance of explicit channel relevance assessment and represents a step toward more robust and effective activation mechanisms in deep learning.

Acknowledgements

This work is supported by the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (Grant No. SN-ZJU-SIAS-001), the Zhejiang Provincial Natural Science Foundation of China (Grant No. LMS26F020019) and the Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

- Ali, A., Schnake, T., Eberle, O., Montavon, G., Müller, K.-R., and Wolf, L. Xai for transformers: Better explanations through conservative propagation. In *International conference on machine learning*, pp. 435–451. PMLR, 2022.
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.
- Bai, Y., Zeng, Y., Jiang, Y., Xia, S.-T., Ma, X., and Wang, Y. Improving adversarial robustness via channel-wise activation suppressing. *arXiv preprint arXiv:2103.08307*, 2021.
- Ballé, J., Laparra, V., and Simoncelli, E. P. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*, 2015.
- Chefer, H., Gur, S., and Wolf, L. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021.
- Cheng, L., Liang, Y., Lu, Y., and Cheung, Y.-m. Gradtoken: Decoupling tokens with class-aware gradient for visual explanation of transformer network. *Neural Networks*, 181:106837, 2025.
- Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.
- Deiseroth, B., Deb, M., Weinbach, S., Brack, M., Schramowski, P., and Kersting, K. Atman: Understanding transformer predictions through memory efficient attention manipulation. *Advances in Neural Information Processing Systems*, 36:63437–63460, 2023.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- Dubey, S. R., Singh, S. K., and Chaudhuri, B. B. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.
- Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., and Garcia, R. Incorporating second-order functional knowledge for better option pricing. *Advances in neural information processing systems*, 13, 2000.
- Feng, Z., Hu, J., Wu, S., Yu, X., Song, J., and Song, M. Model doctor: A simple gradient aggregation strategy for diagnosing and treating cnn classifiers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 616–624, 2022.
- Geva, M., Caciularu, A., Wang, K. R., and Goldberg, Y. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*, 2022.
- Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323. JMLR workshop and conference proceedings, 2011.
- Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., and Wang, Y. Transformer in transformer. *Advances in neural information processing systems*, 34:15908–15919, 2021.
- Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.
- Ivanovs, M., Kadikis, R., and Ozols, K. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, 150:228–234, 2021.
- Kipf, T. and Welling, M. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2016.
- Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.
- Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.
- Liu, A., Hu, H., Qiu, T., Zhou, Q., Guan, Q., and Li, X. Exploring optimal adaptive activation functions for various tasks. In *IEEE international conference on bioinformatics and biomedicine (BIBM)*, pp. 2290–2297. IEEE, 2020.
- Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.
- Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the european conference on computer vision*, pp. 116–131, 2018.
- Mandler, H. and Weigand, B. A review and benchmark of feature importance methods for neural networks. *ACM Computing Surveys*, 56(12):1–30, 2024.
- McClelland, J. L., Rumelhart, D. E., Group, P. R., et al. *Parallel Distributed Processing, Volume 2: Explorations in the Microstructure of Cognition: Psychological and Biological Models*, volume 2. MIT press, 1987.
- Nan, Q., Cao, J., Zhu, Y., Wang, Y., and Li, J. Mdfend: Multi-domain fake news detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3343–3347, 2021.
- Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.
- Panda, M. P., Tiezzi, M., Vilas, M., Roig, G., Eskofier, B. M., and Zanca, D. Fovex: Human-inspired explanations for vision transformers and convolutional neural networks. *International Journal of Computer Vision*, pp. 1–23, 2025.
- Qiu, T., Gao, C., Feng, Z., Lei, J., Hu, B., Wang, X., Gao, Y., and Song, M. Dual-perspective activation: Efficient channel denoising via joint forward-backward criterion for artificial neural networks. *Advances in Neural Information Processing Systems*, 37:70499–70519, 2024a.
- Qiu, T., Xu, W., Chen, L., Zhou, L., Feng, Z., and Song, M. Dynamic neural response tuning. In *The twelfth international conference on learning representations*, 2024b.
- Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021.
- Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.
- Ren, J., Xia, F., Lee, I., Noori Hoshyar, A., and Aggarwal, C. Graph learning for anomaly analytics: Algorithms, applications, and challenges. *ACM transactions on intelligent systems and technology*, 14(2):1–29, 2023.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.
- Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- Tang, J., Li, J., Gao, Z., and Li, J. Rethinking graph neural networks for anomaly detection. In *International conference on machine learning*, pp. 21076–21089. PMLR, 2022.

- Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021a.
- Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 32–42, 2021b.
- Vilas, M. G., Schaumlöffel, T., and Roig, G. Analyzing vision transformers for image classification in class embedding space. *Advances in neural information processing systems*, 36:40030–40041, 2023.
- Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E., et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.
- Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021.
- Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., and Leiserson, C. E. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.
- Wightman, R. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.
- Wu, Z. and Ong, D. C. On explaining your explanations of bert: An empirical study with sequence classification. *arXiv preprint arXiv:2101.00196*, 2021.
- Yan, H., Zhang, J., Niu, G., Feng, J., Tan, V., and Sugiyama, M. Cifs: Improving adversarial robustness of cnns via channel-wise importance-based feature selection. In *International Conference on Machine Learning*, pp. 11693–11703. PMLR, 2021.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.
- Zhou, Y., Zheng, H., Huang, X., Hao, S., Li, D., and Zhao, J. Graph neural networks: Taxonomy, advances, and trends. *ACM transactions on intelligent systems and technology*, 13(1):1–54, 2022.

A. Appendix

A.1. Quantitative Correlation Analysis

Based on Figure 1(b1, b2), denote the vector that records activity levels (b1, blue) as $\mathbf{v}$ and the vector that records decision weights (b2, green) as $\mathbf{w}$. They both contain 768 channels. We provide the following quantitative metrics:

- **Pearson correlation coefficient** between $\mathbf{v}$ and $\mathbf{w}$, which ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), emphasizing a linear relationship.
- **Spearman correlation coefficient** between $\mathbf{v}$ and $\mathbf{w}$, which ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation), emphasizing a monotonic relationship.
- **Average intrinsic score** of $\mathbf{v}$ and **Average extrinsic score** of $\mathbf{w}$ under different channel subsets $\mathbb{S}$. The lower the score, the less relevant the channels in $\mathbb{S}$ are to the task.

The Pearson and Spearman correlations between $\mathbf{v}$ (Figure 1(b1), blue) and $\mathbf{w}$ (Figure 1(b2), green) are shown in Table 7, which shows a considerable positive correlation, providing solid quantitative evidence for the ‘‘intrinsic-extrinsic consensus’’.

Table 7. Correlations between activity levels $\mathbf{v}$ (Figure 1(b1), blue) and decision weights $\mathbf{w}$ (Figure 1(b2), green).

Pair	Pearson Corr.	Spearman Corr.
$\mathbf{v}$ & $\mathbf{w}$	0.5549	0.5202

Moreover, average intrinsic and extrinsic scores under different channel subsets are shown in Table 8, which further demonstrates the high consistency between intrinsic and extrinsic views in identifying irrelevant channels and that identifying irrelevant channels under intrinsic-extrinsic consensus is better than that under a single view or their combination.

Table 8. Average intrinsic and extrinsic scores under different channel subsets.

Channel Subset ($\mathbb{S}$)	Avg. Intrinsic Score $\frac{1}{\ \mathbb{S}\ } \sum \mathbf{v}_{c \in \mathbb{S}}$	Avg. Extrinsic Score $\frac{1}{\ \mathbb{S}\ } \sum \mathbf{w}_{c \in \mathbb{S}}$
All Channels	-0.2813	-0.0237
Irrelevant by Intrinsic Only ($\mathcal{G}^{k, Intr}$, Eq. 9)	-0.3752	-0.0392
Irrelevant by Extrinsic Only ($\mathcal{G}^{k, Extr}$, Eq. 12)	-0.3962	-0.0751
Irrelevant by Combination ($\mathcal{G}^{k, Intr} \cup \mathcal{G}^{k, Extr}$)	-0.3660	-0.0391
Irrelevant by Consensus ($\mathcal{G}^{k, Intr} \cap \mathcal{G}^{k, Extr}$, Eq. 13)	-0.4105	-0.0762
Relevant by Intrinsic Only ($\neg \mathcal{G}^{k, Intr}$, Eq. 9)	+0.2462	+0.0630
Relevant by Extrinsic Only ($\neg \mathcal{G}^{k, Extr}$, Eq. 12)	+0.1702	+0.1326
Relevant by Combination ($\neg \mathcal{G}^{k, Intr} \cup \neg \mathcal{G}^{k, Extr}$)	+0.1972	+0.0690
Relevant by Consensus ($\neg \mathcal{G}^{k, Intr} \cap \neg \mathcal{G}^{k, Extr}$, Eq. 13)	+0.3683	+0.1354

Additionally, the proportions of active channels ($\mathbf{v}_i > 0$) with (Figure 3(a1)) and without (Figure 3(b1)) the proposed AIEC are shown in Table 9, which implies that irrelevant channels are effectively identified and suppressed, leading to notable sparsity of the feature representation.

Table 9. Proportions of active channels with (Figure 3(a1)) and without (Figure 3(b1)) the proposed AIEC.

Figure 3	w/o AIEC (a1)	w/ AIEC (b1)
Proportion of Active Channels	14.84%	7.68%

A.2. Hyperparameter Impact Analysis

Activation threshold τ for the Threshold Activation Unit (TAU). To achieve preliminary sparsity, only input signals that exceed the threshold τ are activated, while those below the threshold are inhibited to zero. Therefore, τ is a non-negative value. Figure 4 demonstrates that as τ increases, the performance decreases. Possible reasons could be the influence of weight initialization and feature normalization operations. Typically, weights are initialized using a distribution with a

mean of zero, and normalization techniques such as layer normalization and batch normalization are used to make the feature distribution centered around zero (by subtracting the feature mean) to eliminate shifts in data covariates. Under these circumstances, $\tau = 0$ becomes the optimal activation threshold. Additionally, it may be feasible to modify the strategy for weight initialization and feature normalization to achieve optimal effects when considering a positive τ .

Balancing parameter λ_{gate} for the gatekeeping loss $\mathcal{L}_{gate}$. The optimal λ_{gate} for $\mathcal{L}_{gate}$ is specific to individual tasks. The relationship between λ_{gate} and the accuracy on CIFAR-100 with ViT-Tiny is depicted in Figure 5. In this case, the optimal λ_{gate} is roughly 200, and too large a λ_{gate} can result in negative side effects. For other trials, we found the optimal λ_{gate} to be 400 and 200 respectively when training DeiT-Tiny and TNT-Small on CIFAR-100 and the optimal λ_{gate} to be 200 when training ViT-Tiny on CIFAR-10. Searching for the optimal λ_{gate} for each task is time-consuming. Therefore, for the majority of our experiments, we set the default λ_{gate} to 200.

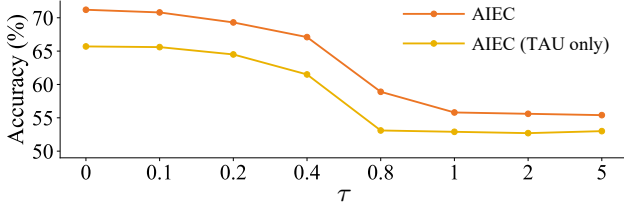


Figure 4. Top-1 accuracy (%) with respect to the activation threshold τ for the Threshold Activation Unit (TAU) when training on CIFAR-100 with ViT-Tiny.

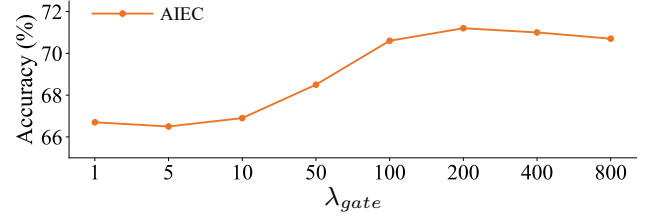


Figure 5. Top-1 accuracy (%) with respect to the balancing parameter λ_{gate} for the gatekeeping loss $\mathcal{L}_{gate}$ when training on CIFAR-100 with ViT-Tiny.

A.3. Computational Complexity

Theoretically, the time complexity and space complexity of AIEC’s Threshold Activation Unit (TAU), Activation-Counting Unit (ACU), Decision-Making Unit (DMU), and Consensus Gatekeeping Unit (CGU) are shown in Table 10,

Table 10. Training computational complexity of the proposed AIEC.

AIEC	TAU	ACU	DMU	CGU
Training Time Complexity	$\mathcal{O}(Input)$	$\mathcal{O}(C)$	$\mathcal{O}(C \cdot K)$	$\mathcal{O}(C)$
Training Space Complexity	$\mathcal{O}(Input)$	$\mathcal{O}(C \cdot K)$	$\mathcal{O}(C \cdot K)$	$\mathcal{O}(C)$
Inference Time Complexity	$\mathcal{O}(Input)$	not involved	not involved	not involved
Inference Space Complexity	$\mathcal{O}(Input)$	not involved	not involved	not involved

where $Input$ denotes the shape of the input feature map, C denotes the number of feature channels, K denotes the number of classes. Note that DMU’s computation comes from matrix multiplication (Eq. 10) and loss computation, in which the matrix multiplication (Eq. 10) can be highly accelerated through parallel processing. Therefore in practice, although the computational complexity of DMU is $\mathcal{O}(C \cdot K)$ while ACU is $\mathcal{O}(C)$, the actual time cost of DMU is much smaller than that of ACU. The average time taken by each part of AIEC in the ViT-Tiny model (the sum of 12 layers) to process a batch of data (1024 images with 224×224-pixels) on an NVIDIA A6000 GPU is shown in Table 11.

Table 11. Average time taken by each part of the proposed AIEC to process a batch of data.

AIEC	TAU	ACU	DMU	CGU
Training Time per Batch	3.282ms	100.303ms	4.706ms	12.048ms

ACU actually has the highest time overhead, while DMU has the second lowest, even close to the fastest TAU. In addition, we can further reduce the time cost of ACU (Eq. 4-8) and DMU (Eq. 10-11) by lowering their update frequency (e.g., updating every 5 iterations).

A.4. Training and Inference Procedures

This section provides a comprehensive description of the training and inference procedures as elucidated in Algorithm 1 and 2. During the training phase, TAU, ACU, DMU, and CGU work together. The responses from irrelevant channels are

suppressed under the guidance of the gatekeeping loss $\mathcal{L}_{gate}$ (Eq. 15). By the end of training, the model has already learned how to avoid generating responses on irrelevant channels. Hence, during the inference phase, ACU, DMU, and CGU are all discarded, and only TAU needs to be involved.

Algorithm 1 : Training procedure

Input: $\mathbb{D}_{train}^K$: Training dataset with K categories; T : Maximum number of iterations; l : Layer index where AIEC is applied (one layer as an example here; multiple layers are possible).

Initialize: $f_{1:L}(\cdot)$: Model with L layers; $\{\theta^k, \eta^k, \phi^k\}_{k=1}^K$: ACU's counters; $\{\mathbf{w}^k\}_{k=1}^K$: DMU's decision weights; $\lambda_{aux}, \lambda_{gate}$: Balacing parameters.

- 1: **for** $t = 1, 2, \dots, T$ **do**
 - 2: Sample $\{\mathbf{X}^k, y^k\}$ from $\mathbb{D}_{train}^K$, which is of category k .
 - 3: Compute pre-activation feature map $\mathbf{F}^k = f_{1:l}(\mathbf{X}^k)$.
 - 4: Compute post-activation feature map $\mathbf{A}^k = \text{TAU}(\mathbf{F}^k)$, whose elements are all non-negative.
 - 5: Compute post-activation feature vector $\mathbf{a}^k = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W \mathbf{A}_{h,w}^k$, whose elements are all non-negative.
 - 6: Binarize $\mathbf{a}^k$ to $\tilde{\mathbf{a}}_c^k = \begin{cases} 1, & \text{if } \mathbf{a}_c^k > 0 \\ 0, & \text{if } \mathbf{a}_c^k \leq 0 \end{cases}$ based on whether a channel is activated or not.
 - 7: Update counters $\theta^{k,t} = \theta^{k,t-1} + \tilde{\mathbf{a}}_c^k$, $\eta^{k,t} = \eta^{k,t-1} + (1 - \tilde{\mathbf{a}}_c^k)$, $\phi^{k,t} = \phi^{k,t-1} + 1$.
 - 8: Compute "relative firing rate" $\mathbf{v}^{k,t} = \frac{\theta^{k,t} - \eta^{k,t}}{\phi^{k,t}}$.
 - 9: Establish intrinsic channel relevance division $\mathcal{G}_c^{k, Intr} = \begin{cases} 0, & \text{if } \mathbf{v}_c^k > 0 \\ 1, & \text{if } \mathbf{v}_c^k \leq 0 \end{cases}$.
 - 10: Obtain intermediate predictions $\{p^k\}_{k=1}^K = \text{softmax}(\mathbf{a}^k \cdot \{\mathbf{w}^k\}_{k=1}^K)$.
 - 11: Compute auxiliary loss $\mathcal{L}_{aux} = -\sum_{k=1}^K y^k \log(p^k)$ for learning $\{\mathbf{w}^k\}_{k=1}^K$.
 - 12: Establish extrinsic channel relevance division $\mathcal{G}_c^{k, Extr} = \begin{cases} 0, & \text{if } \mathbf{w}_c^k > 0 \\ 1, & \text{if } \mathbf{w}_c^k \leq 0 \end{cases}$.
 - 13: Establish final channel relevance division $\{\mathcal{G}^k\}_{k=1}^K = \{\mathcal{G}_c^{k, Intr} \cap \mathcal{G}_c^{k, Extr}\}_{k=1}^K$ based on intrinsic-extrinsic consensus, where the indicator "1" signifies potential irrelevant channels.
 - 14: Filter out irrelevant channels $\tilde{\mathbf{a}}^k = \mathbf{a}^k \odot \mathcal{G}^k$.
 - 15: Impose gatekeeping on irrelevant channels by loss $\mathcal{L}_{gate} = \frac{\sum_{c=1}^C \|\tilde{\mathbf{a}}_c^k\|_2}{\sum_{c=1}^C \mathcal{G}_c^k}$.
 - 16: Keep passing $\mathbf{A}^k$ forward until the final layer outputs $\mathbf{logits} = f_{l+1:L}(\mathbf{A}^k)$.
 - 17: Compute task-level loss $\mathcal{L}_{task} = \text{CrossEntropy}(\mathbf{logits}, y^k)$.
 - 18: Compute the final loss $\mathcal{L} = \mathcal{L}_{task} + \lambda_{aux} \cdot \mathcal{L}_{aux} + \lambda_{gate} \cdot \mathcal{L}_{gate}$.
 - 19: Update model parameters through back-propagation and gradient descent.
 - 20: **end for**
-

Algorithm 2 : Inference procedure

Input: $\mathbf{X}$: a test sample with its category unknown.

Initialize: $f_{1:L}(\cdot)$: Trained model with L layers.

- 1: Compute pre-activation feature map $\mathbf{F} = f_{1:l}(\mathbf{X})$.
 - 2: Compute post-activation feature map $\mathbf{A} = \text{TAU}(\mathbf{F})$.
 - 3: Keep passing $\mathbf{A}$ forward until the final layer outputs $\mathbf{logits} = f_{l+1:L}(\mathbf{A})$.
 - 4: Predict $\mathbf{X}$'s label $\hat{y} = \text{argmax}(\mathbf{logits})$.
-

A.5. Discussion on Linear Decision Weights

In Decision-Making Unit (DMU) of the proposed AIEC, linear decision weights without the bias term are chosen to quantify each channel’s contribution/relevance to the network’s prediction/decision. The key reason for this choice is that: Linear models are highly interpretable, which can be used with positive (or non-negative) inputs for interpretability analysis.

The number of channels (feature vector’s length) is C . In Decision-Making Unit (DMU), the predicted score y is typically obtained through the dot product of the input features $\mathbf{x} = \{x_1, x_2, \dots, x_C\}$ (i.e., the $\mathbf{a}$ in Eq. 3 and 10) and linear weights $\mathbf{w} = \{w_1, w_2, \dots, w_C\}$ as $y = \mathbf{x} \cdot \mathbf{w}^T$ (Eq. 10). This process can be expressed in the form of weighted summation as $y = w_1x_1 + w_2x_2 + \dots + w_Cx_C$. With positive input feature x_c ($x_c > 0$), its contribution w_cx_c to the score y exhibits a positive correlation with its weight w_c . To get a higher score y , important features tend to be assigned higher weights to amplify their impact. Consequently, the learned linear weights $\{w_1, w_2, \dots, w_C\}$ can quantify the relevance of input features $\{x_1, x_2, \dots, x_C\}$ to the network’s decision in a one-to-one correspondence (i.e., w_c to x_c). From a purely symbolic perspective (Eq. 12), a positive weight $w_c > 0$ implies that the feature x_c contributes positively to the predicted score y , while a negative weight $w_c < 0$ indicates that the feature x_c has a negative contribution to the predicted score y , in which case suppressing x_c to zero would increase y (Eq. 15, it is another explanation of the gatekeeping).

In contrast, more complex classifiers such as MLPs or non-linear heads do not preserve such interpretability and are more like “black boxes”. Therefore, what we need is not a highly expressive classifier, but rather an interpretable one, whose weights can reflect the relevance of input features to the network’s decision in a one-to-one correspondence (i.e., w_c to x_c).

Moreover, since the weights must correspond one-to-one with the elements of the input feature vector (i.e., w_c to x_c), the length of weights must be C to match the feature vector’s length, and also since the channel relevance assessment is category-specific that requires distinct weights to be learned for each category (K categories in total), the weights of shape $K \times C$ is the only option, which is a linear classifier. Alternatives such as MLPs or non-linear heads are not applicable in this context. Even if they were applicable, their non-linear factors would make the conclusion less reliable.

A.6. Differences between Channel Gatekeeping and Channel Pruning

The proposed channel gatekeeping method differs fundamentally from channel pruning in terms of objective, mechanism, and implementation, as detailed below.

- The target of channel pruning is channel weights $\mathbf{W}$, while the target of channel gatekeeping is channel responses $f(\text{input}; \mathbf{W})$. Channel pruning operates by setting **the weights of certain channels** to zero, resulting in no responses on pruned channels for any input (This is why channel pruning typically leads to accuracy decrease.). In contrast, channel gatekeeping operates by suppressing **the responses of certain channels** to zero, in which case, the channel weights are not necessarily zero, and the suppressed channels can vary for different inputs.
- Channel pruning requires post-processing to remove irrelevant channel weights and some need further fine-tuning, while channel gatekeeping is conveniently trained end-to-end from scratch and does not require any post-processing or fine-tuning.
- The objective of channel pruning is trying to reduce computation and storage requirements without sacrificing accuracy, while the objective of channel gatekeeping is trying to improve accuracy without increasing computational overhead. The objectives of the two methods are completely opposite.

A.7. More Visualization Results

This section extends §4.4 of the main paper, providing visualizations for all categories in CIFAR-10, as shown in Figure 6, 7, 8, 9. With the proposed AIEC, the activation responses become sparser, as evident from the activity levels across channels. This implies that AIEC has identified and suppressed some irrelevant channels. Furthermore, channel gatekeeping in the AIEC results in sharper decision weight boundaries, demonstrating that the network gains greater confidence in prioritizing channels that consistently support correct decisions. These suggest that the AIEC enhances the precision of key feature extraction, reducing learning difficulty and improving model interpretability. Notably, more than half of the channels exhibit their decision weights close to zero, which corresponds to their activity levels dropping to a minimum, meaning these channels are completely inhibited. As a result, the decision weights learn no useful information from them. In contrast, the other active channels provide meaningful input to the decision weights, allowing the weights to assess each channel’s contribution/relevance and reflect it in the sign and magnitude of the weights.

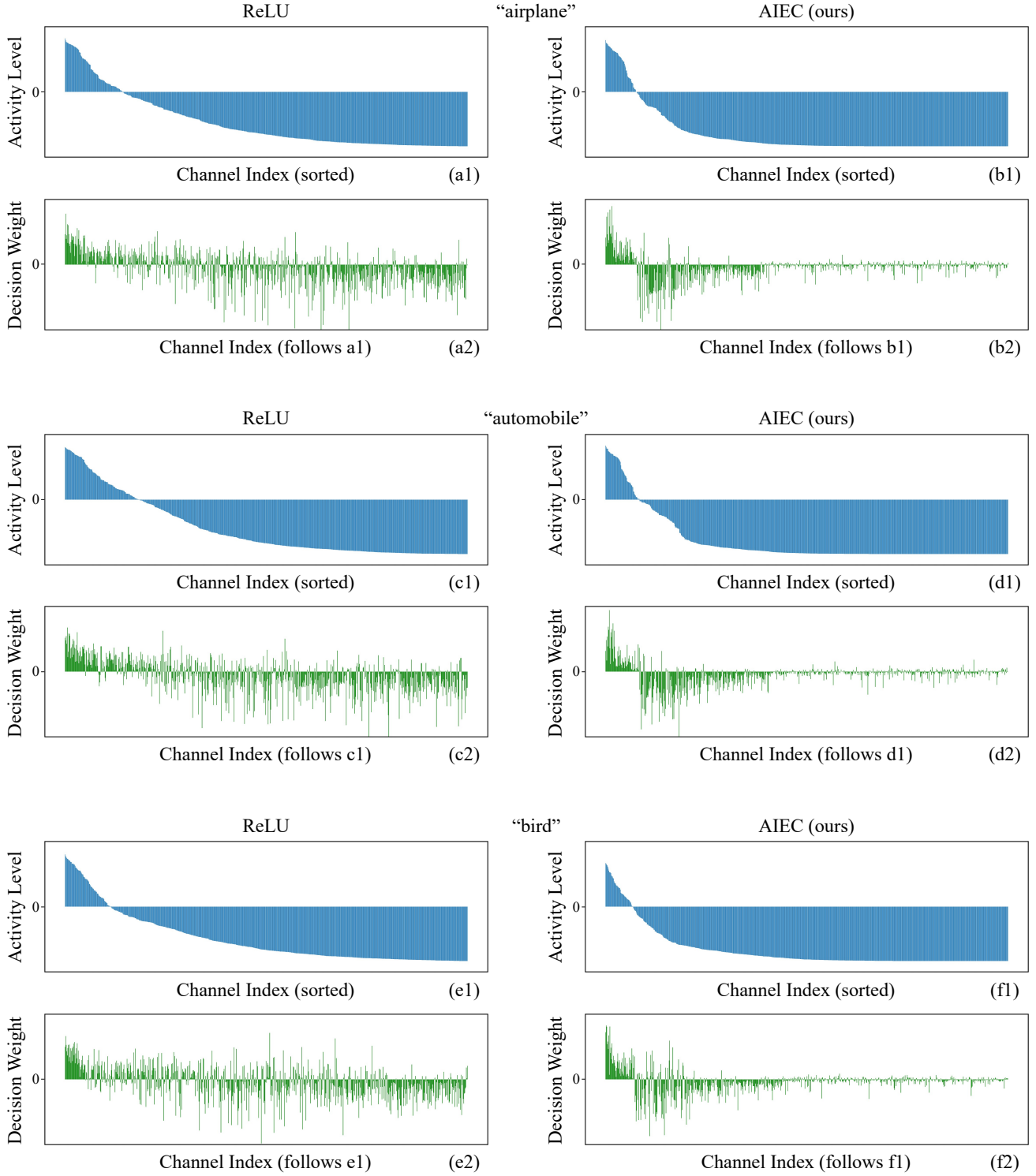


Figure 6. More visualization results extending Figure 3 of the main paper, specifically for the "airplane", "automobile", and "bird" categories in the CIFAR-10 dataset.

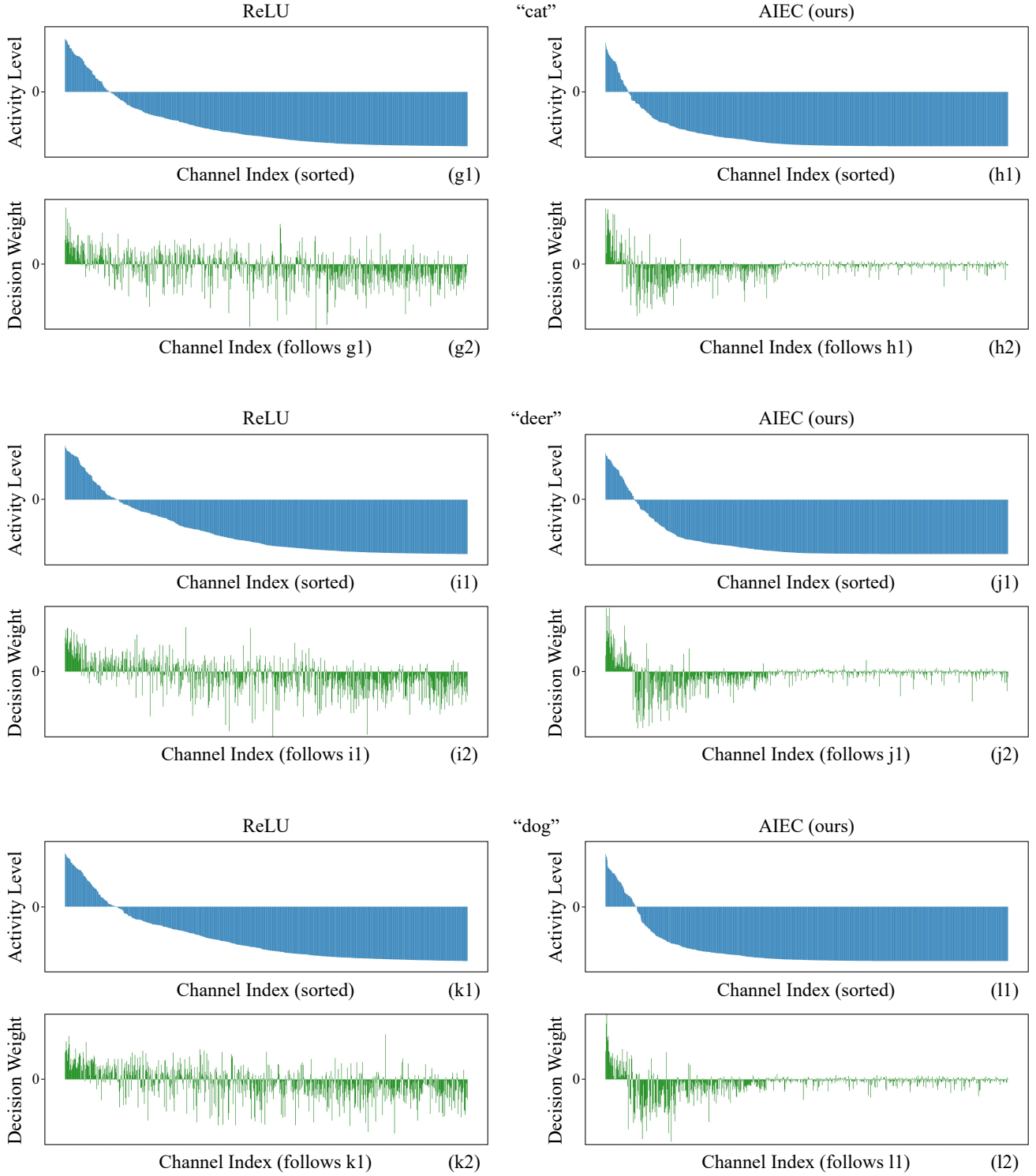


Figure 7. More visualization results extending Figure 3 of the main paper, specifically for the "cat", "deer", and "dog" categories in the CIFAR-10 dataset.

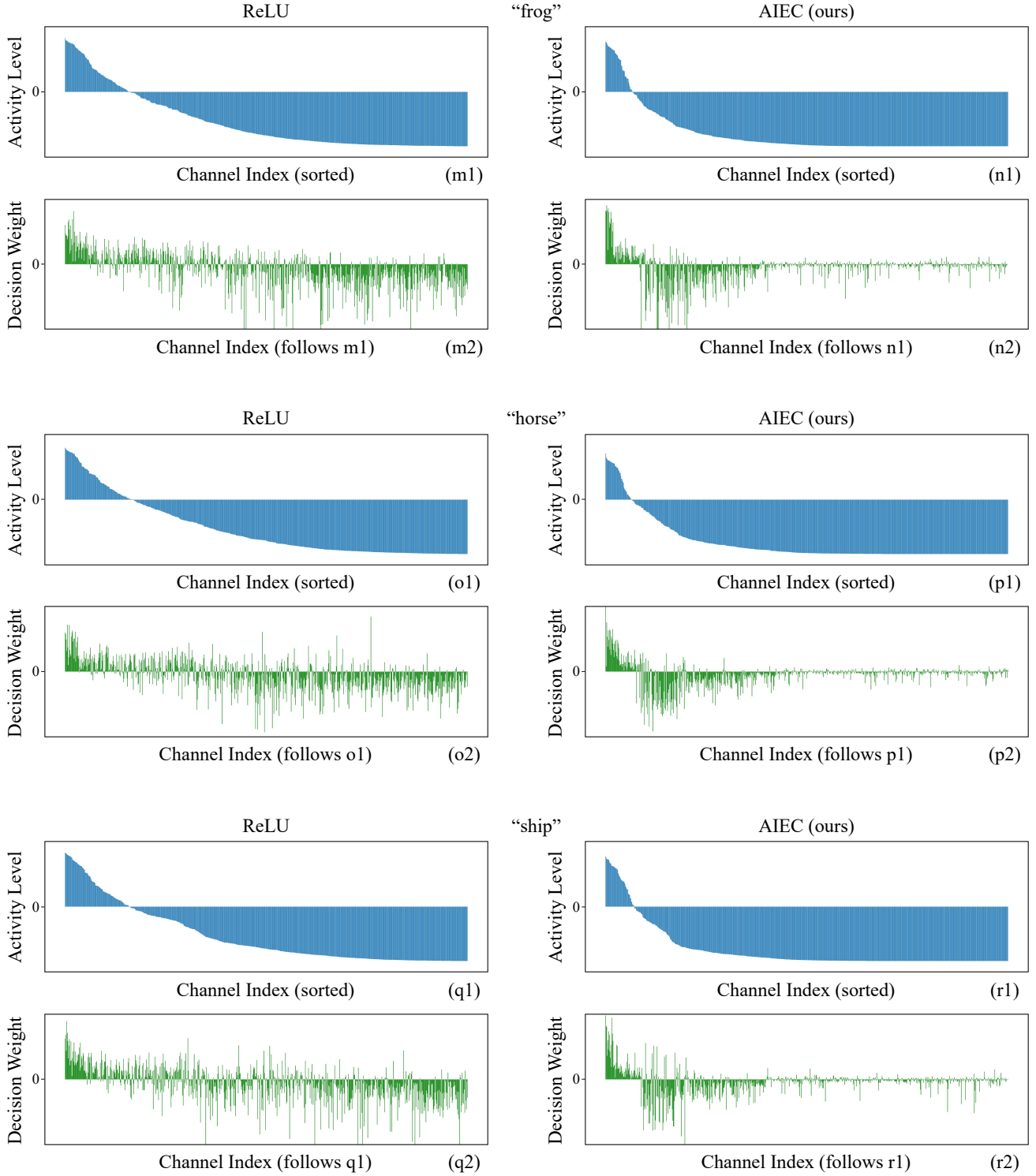


Figure 8. More visualization results extending Figure 3 of the main paper, specifically for the "frog", "horse", and "ship" categories in the CIFAR-10 dataset.

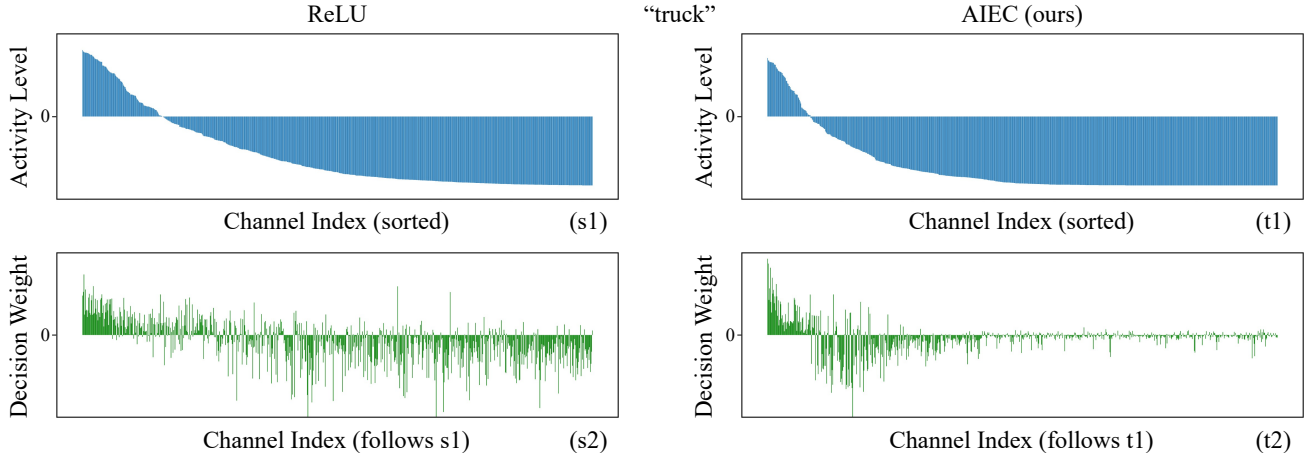


Figure 9. More visualization results extending Figure 3 of the main paper, specifically for the “truck” category in the CIFAR-10 dataset.

A.8. Detailed Experimental Settings

Table 12. Experimental settings. “(left) / (right)” is for CIFAR-{10,100} (left) and ImageNet-{100,1K} (right), respectively.

Argument	Value
epochs	300
batch size	identical in each comparison group
optimizer	adamw
learning rate	$0.0005 \times (\text{batch_size} / 512)$
learning rate scheduler	cosine annealing
weight decay	$5e-2$
warmup epochs	$5 / 20$
warmup learning rate	$1e-6$
warmup scheduler	linear
dropout rate	0.0
drop path rate	0.1
gradient clipping norm	1.0
image size	$32 \times 32 / 224 \times 224$
horizontal flipping rate	0.5
vertical flipping rate	0.0
color jitter	0.4
auto augment	rand-m9-mstd0.5-inc1 (from “timm”)
interpolation	random
random erasing prob	0.25
random erasing mode	pixel
random erasing count	1
training from scratch (no pre-training)	✓
automatic mixed precision training	✓
τ	0
$\mathcal{L}_{aux}$	1
$\mathcal{L}_{gate}$	depending on networks and datasets