Winsor-CAM: Human-Tunable Visual Explanations from Deep Networks via Layer-Wise Winsorization

Casey Wall, Longwei Wang, Rodrigue Rizk, KC Santosh

***Abstract*—Interpreting the decision-making process of Convolutional Neural Networks (CNNs) is critical for deploying models in high-stakes domains. Gradient-weighted Class Activation Mapping (Grad-CAM) is a widely used method for visual explanations, yet it typically focuses on the final convolutional layer or naïvely averages across layers, strategies that can obscure important semantic cues or amplify irrelevant noise. We propose Winsor-CAM, a novel, human-tunable extension of Grad-CAM that generates robust and coherent saliency maps by aggregating information across all convolutional layers. To mitigate the influence of noisy or extreme attribution values, Winsor-CAM applies Winsorization, a percentile-based outlier attenuation technique. A user-controllable threshold allows for semantic-level tuning, enabling flexible exploration of model behavior across representational hierarchies. Evaluations on standard architectures (ResNet50, DenseNet121, VGG16, InceptionV3) using the PASCAL VOC 2012 dataset demonstrate that Winsor-CAM produces more interpretable heatmaps and achieves superior performance in localization metrics, including intersection-over-union and center-of-mass alignment, when compared to Grad-CAM and uniform layer-averaging baselines. Winsor-CAM advances the goal of trustworthy AI by offering interpretable, multi-layer insights with human-in-the-loop control.**

***Impact Statement*—Winsor-CAM enhances transparency in deep learning by enabling expert-guided visual explanations across all convolutional layers, helping to address interpretability challenges in high-stakes domains. The method enables interactive exploration of saliency across semantic levels, supporting more transparent AI-assisted decision-making. Winsor-CAM is compatible with standard CNN architectures and requires no model retraining or architectural modifications, making it suitable for integration into existing interpretability pipelines. As a post-hoc explainability tool, Winsor-CAM is most effective when applied to well-trained models and interpreted by domain experts, reinforcing the importance of responsible AI deployment. This work contributes toward trustworthy and interpretable AI by enabling human-centered, transparent model assessment.**

***Index Terms*—Explainable AI (XAI), Convolutional Neural Networks (CNNs), Winsorization, Saliency Maps.**

I. Introduction

Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance across a variety of computer vision tasks, including image classification, object detection, and medical imaging. Despite these advances, the decision-making processes of CNNs remain largely opaque, raising concerns about accountability, reliability, and public trust, particularly in high-stakes domains such as healthcare, autonomous systems, and law enforcement. As a result, the field of Explainable Artificial Intelligence (XAI) has emerged with the goal of developing techniques that make neural network predictions more interpretable and transparent to human users [1], [2].

Among the growing suite of XAI techniques, visual explanation methods have become especially prominent due to their intuitive appeal and applicability to spatially structured inputs such as images [3], [4]. These methods

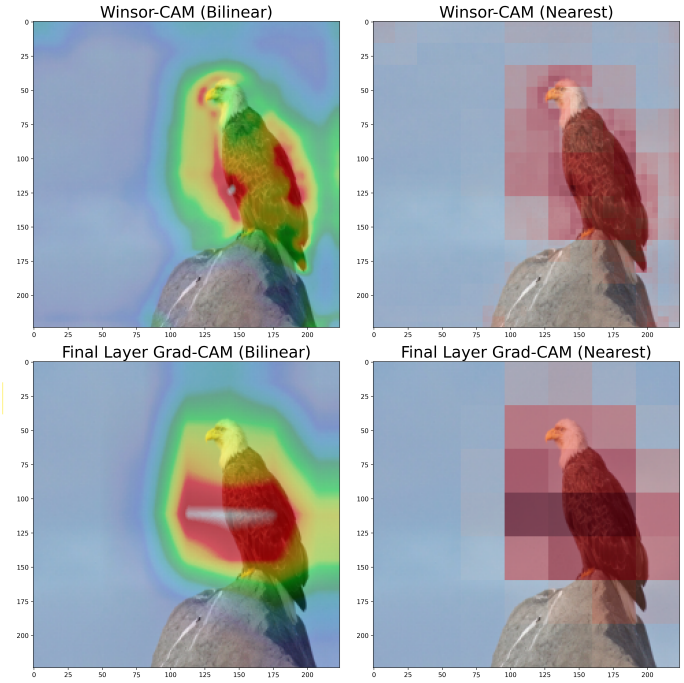

Fig. 1. Comparison of Winsor-CAM and standard Grad-CAM outputs on a ResNet-50 model, illustrating improved localization and robustness to interpolation artifacts. Winsor-CAM produces smoother, semantically aligned heatmaps under both bilinear and nearest-neighbor upsampling, while Grad-CAM exhibits spatial distortion and noise, particularly under nearest interpolation.

Code will be publicly available at time of publication.

IEEE Transactions on Pattern Analysis and Machine Intelligence

C. Wall, L. Wang, R. Rizk, and KC Santosh are with the Artificial Intelligence Research Lab, Department of Computer Science, University of South Dakota (e-mail: casey.wall@coyotes.usd.edu, {longwei.wang, rodrigue.rizk, kc.santosh}@usd.edu).

generate saliency maps or heatmaps that highlight the most influential regions of an input image relevant to a model's prediction, offering users visual insight into the model's focus during inference. An example of generated heatmaps can be found in Fig. 1. This capability supports a range of downstream objectives, from error diagnosis and model debugging to fairness auditing and scientific discovery [5]–[9].

A seminal approach in this space is Class Activation Mapping (CAM), introduced by Zhou et al. [10], which localizes discriminative regions by projecting classifier weights onto convolutional feature maps. CAM exploits the hierarchical and spatial structure of CNNs to trace decision-relevant regions at the final convolutional layer. Despite its effectiveness, CAM requires architectural modifications and is thus limited in flexibility. Subsequent developments, such as Gradient-weighted Class Activation Mapping (Grad-CAM) [11], addressed this limitation by leveraging gradients to compute class-specific importance scores, enabling visual explanations without modifying the underlying model architecture. Grad-CAM has since become a de facto standard for visual explanation due to its generality and ease of use in CNN-based XAI.

However, Grad-CAM suffers from a key limitation: it derives explanations from only the final convolutional layer. While this layer captures high-level semantic features, it may overlook low-level cues, such as textures or edges, learned in earlier layers. Furthermore, naïve extensions that uniformly average Grad-CAM outputs across layers can dilute semantically meaningful patterns by introducing noise from less relevant feature maps. These limitations motivate the need for explanation methods that incorporate multi-layer information while mitigating inter-layer variance and suppressing outlier dominance.

In this paper, we propose **Winsor-CAM**, a novel extension of Grad-CAM designed to overcome these limitations by leveraging saliency information from all convolutional layers in a CNN. The key innovation in Winsor-CAM is the integration of *Winsorization*, a statistical clipping technique that suppresses extreme layer importance values. This technique enables semantic control via a user-adjustable percentile threshold, allowing users to dynamically adjust the semantic resolution of the output.

Our contributions are as follows:

1) We present Winsor-CAM, the first method to aggregate Grad-CAM explanations across the entire convolutional stack while applying robust outlier attenuation via Winsorization.
2) We introduce a human-controllable percentile parameter to tune the semantic abstraction level of the explanations.
3) We provide a comprehensive evaluation across standard CNN architectures and demonstrate improved interpretability and localization fidelity.
4) We show that Winsor-CAM outperforms commonly used baselines like final layer Grad-CAM, naïve layer aggregation, Grad-CAM++, LayerCAM, and ShapleyCAM in terms of visual coherence and spatial alignment with ground truth segmentation masks.
5) We establish Winsor-CAM as a paradigm shift in CNN interpretability, redefining saliency as a tunable, hierarchical process rather than a fixed, single-layer attribution.

To validate our method, we conduct comprehensive experiments using four canonical CNN architectures—ResNet50, DenseNet121, VGG16, and InceptionV3—on a subset of the PASCAL VOC 2012 dataset. We evaluate Winsor-CAM both qualitatively and quantitatively using Intersection over Union (IoU) and the Euclidean distance between the Center-of-Mass (CoM) of the saliency map and the ground truth segmentation mask as proxies for localization accuracy. Our results show that Winsor-CAM consistently outperforms final layer Grad-CAM and naïve layer aggregation in terms of visual interpretability and spatial localization. Moreover, the tunable percentile parameter enables targeted exploration of feature hierarchies, allowing users to emphasize low- or high-level representations as needed. This flexibility makes Winsor-CAM particularly well-suited for expert-in-the-loop interpretability, diagnostic decision support, and interactive explainability workflows.

II. Literature Review

A. Background

Visual explanation techniques for deep neural networks, especially CNNs, have become essential tools in the field of XAI. Among these, methods that produce spatially localized visualizations, like saliency maps, identify the regions of an input image most relevant to a model's prediction. These interpretability techniques can broadly be categorized into two classes: **gradient-based** and **perturbation-based**. This taxonomy reflects two fundamental philosophies in XAI: leveraging internal model signals such as gradients and activations (white-box), versus externally probing the model via input perturbations (black-box). Together, these approaches span a wide spectrum of practical strategies for interpreting CNN decisions in complex vision tasks [11]–[18], with some methods combining both perspectives [19], [20].

B. Related Work

1) Gradient-Based Methods: Gradient-based methods rely on computing the derivatives of the model output with respect to internal activations or input features. These gradients are used to infer which neurons or spatial regions most influence the model's decision. Because they operate within the architecture, these methods assume full (white-box) access to model internals and typically offer high-resolution localization and computational efficiency.

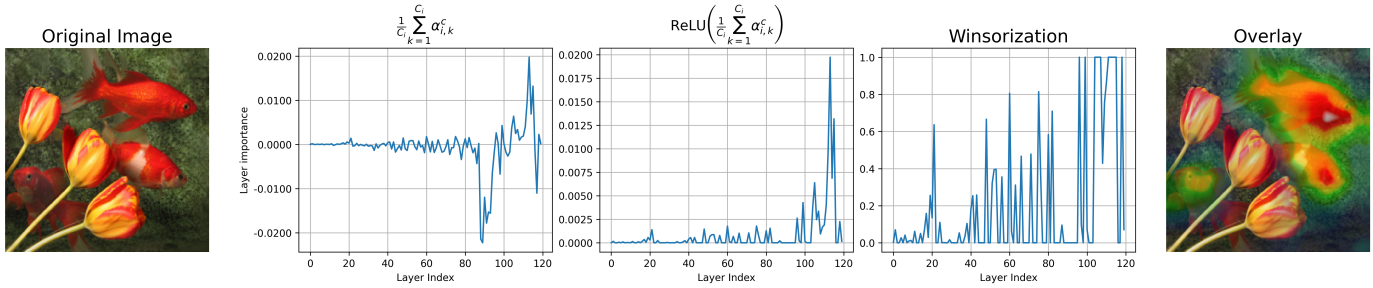

Fig. 2. Step-by-step visualization of the Winsor-CAM pipeline focusing on layer-wise mean importance. From left to right: the input image, raw layer-wise importance scores (before ReLU), positive importance scores after ReLU activation Γ_i^c (as defined in Eq. (5)), Winsorized importance values on a normalized scale (as defined in Steps 4 and 5), and the final overlay of the aggregated Winsor-CAM heatmap on the input image. This progression illustrates how layer-wise importance is extracted, how outliers are suppressed via Winsorization, and how the general structure of importance is preserved throughout the process.

The most influential of these is Grad-CAM [11], which uses gradients from the final convolutional layer to weight the importance of feature maps and generate class-specific saliency maps. Grad-CAM is highly scalable and compatible with most CNN architectures, making it a widely adopted default for CNN explainability. However, it suffers from a reliance on the final layer, potentially missing important information captured in earlier layers, and is sensitive to gradient noise, especially when gradients are sparse or unstable.

Several variants have emerged to address these limitations. Integrated Gradients (IG) [21] computes attributions by integrating gradients along a straight-line path from a baseline input to the actual input. This method satisfies desirable axioms such as sensitivity and implementation invariance, but it requires careful baseline selection and lacks an intuitive tuning mechanism.

Deep Learning Important FeaTures (DeepLIFT) [8] proposes an alternative by comparing activations to those of a reference input and propagating contribution scores through the network. While more efficient than path-based methods, its dependence on a stable reference and assumptions about activation behavior in deeper layers may reduce its robustness.

Layer-wise Relevance Propagation (LRP) [22] redistributes prediction scores backward through the network according to layer-specific rules that preserve relevance conservation. Although LRP offers fine-grained interpretability and full-network attribution, it often requires architectural modifications or specialized propagation rules that limit its general applicability.

Other advanced variants, such as Grad-CAM++ [12] and LayerCAM [13], aim to improve spatial resolution and localization fidelity over the original Grad-CAM formulation. Grad-CAM++ incorporates second-order gradient terms to better handle multi-object scenes and overlapping features. In contrast, LayerCAM introduces a per-pixel saliency mechanism by removing global pooling and applying element-wise gradient-activation products, enabling sharp, layer-specific attributions even at intermediate convolutional depths. This modification allows LayerCAM to localize finer visual concepts at individual layers compared to Grad-CAM, though it does not aggregate saliency across the network hierarchy.

In recent years, several methods have extended Grad-CAM through alternative theoretical frameworks aimed at improving attribution fidelity. Expected Grad-CAM [23] incorporates multiple baselines and kernel smoothing, drawing on ideas from Integrated Gradients to generate smoother, class-specific maps. Shapley-CAM [18] approximates spatial Shapley values using first- and second-order gradients, providing a closed-form alternative to perturbation-based sampling. Axiom-based Grad-CAM [14] offers a principled reformulation by enforcing attribution axioms such as sensitivity and implementation invariance.

Nonetheless, most gradient-based methods remain sensitive to outliers in activation maps and gradient values, and often lack mechanisms for human control over the semantic granularity of the resulting explanations.

2) *Perturbation-Based Methods:* In contrast to gradient-based techniques, perturbation-based methods treat the model as a black box and assess feature importance by measuring changes in model output due to localized alterations in the input. These approaches do not require internal gradients or model architecture knowledge, making them broadly applicable across different model types.

Occlusion Sensitivity [4] is one of the earliest methods in this category. It measures output degradation as small patches of the input image are systematically masked or occluded. Despite its intuitive formulation, it is computationally expensive and limited in semantic resolution, as it cannot easily distinguish between overlapping or distributed features.

Local Interpretable Model-agnostic Explanations (LIME) [24] generates interpretable surrogate models (e.g., linear regressors) by sampling perturbed instances around the input and fitting explanations locally. LIME is flexible and model-agnostic, but its reliance on sampling introduces variance, and it lacks transparency into the model's internal hierarchy of representations.

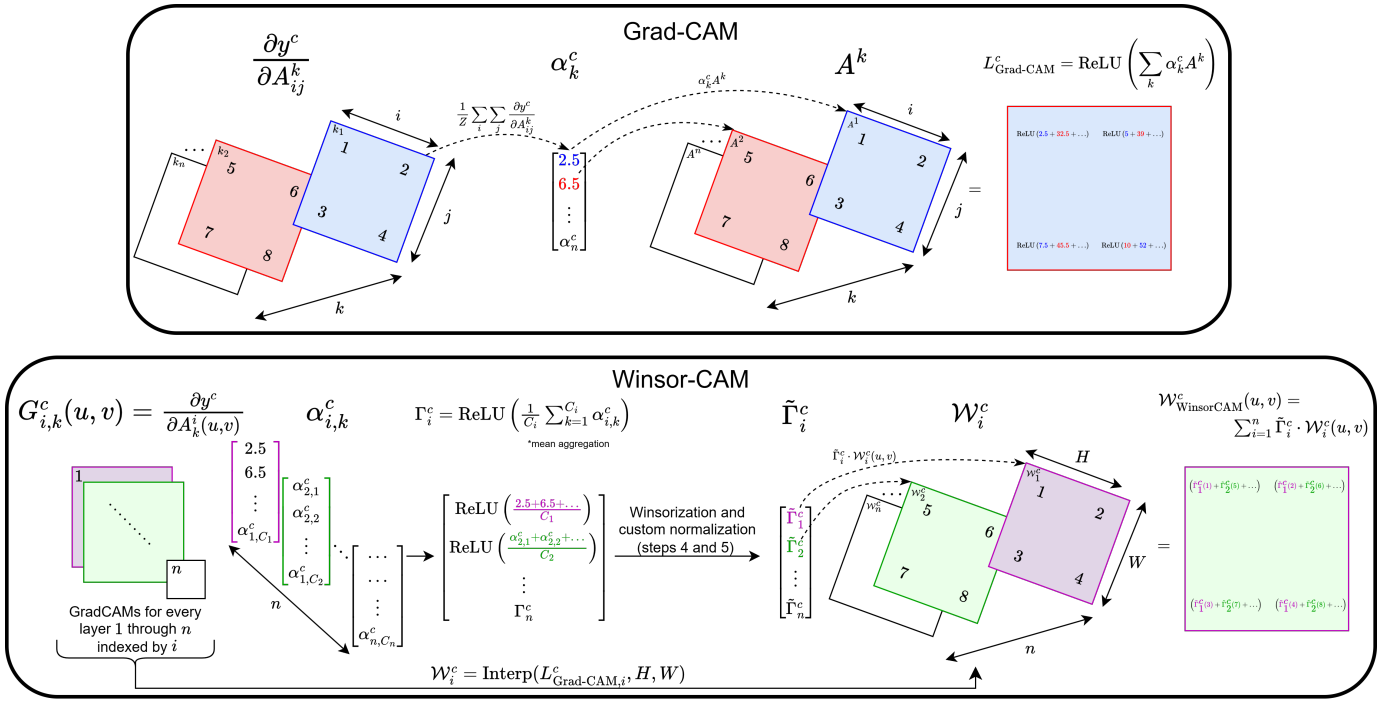

Fig. 3. Comparison of the Grad-CAM and Winsor-CAM pipelines. **Top:** The standard Grad-CAM process applied to a single convolutional layer. Gradients with respect to a target class are computed and average pooled to obtain filter-wise importance weights, which are then used to linearly combine activation maps, producing a heatmap with the same spatial resolution as the selected layer's output. **Bottom:** The Winsor-CAM pipeline. Grad-CAM maps are computed for all convolutional layers, and their corresponding importance weights are used to calculate layer-wise importance scores. These scores are then Winsorized to suppress outliers and normalized (both operations accounting for zero-valued layers, as illustrated in Fig. 2). The resulting normalized importance scores are then used to compute a weighted linear combination of the interpolated Grad-CAM maps from all layers, producing the final high-resolution heatmap.

SHapley Additive exPlanations (SHAP) [25] extends LIME with a game-theoretic framework grounded in Shapley value theory. It provides fair attribution scores with strong theoretical guarantees. However, its computational overhead is significant, especially for deep models, and it primarily focuses on input-level features, offering limited insight into internal spatial hierarchies in CNNs.

Randomized Input Sampling for Explanation (RISE) [26] offers a randomized masking strategy by correlating model predictions with random binary masks applied to the input. This allows for visual saliency estimation without accessing model internals. Although it produces high-quality heatmaps, RISE does not support layer-wise attribution and suffers from randomness-induced instability unless heavily sampled.

In a more recent work, Collective Attribution [17] leverages Shapley value theory (like SHAP) to estimate the collective contribution of individual pixels to object detection outputs, providing class- and box-specific explanations for segmentation tasks. This method, while effective for object detection, suffers from the same computational challenges as other perturbation-based methods, particularly due to high computational costs associated with sampling and perturbation.

3) *Hybrid Methods:* Hybrid methods combine gradient-based sensitivity with perturbation-based averaging to reduce noise in attribution maps while retaining model-specific information. For example, SmoothGrad [19] averages gradients over multiple noisy copies of the input, introducing a perturbation step into otherwise efficient gradient computation. Similarly, Smooth Grad-CAM++ [20] applies the same idea to Grad-CAM++ by averaging its gradient-weighted activation maps across noisy inputs. These methods are generally more computationally expensive than single-pass gradient methods due to multiple forward and backward passes, but remain significantly more efficient than full perturbation-based approaches that require model evaluation on large numbers of masked inputs.

4) *Winsorization for Gradient Stability:* Despite their differences, both gradient-based and perturbation-based methods are susceptible to instability due to noisy activations or unbounded gradients, especially in deep networks. Winsorization, a statistical technique for bounding outliers, improves gradient stability and interpretability by attenuating dominant layer responses. We apply this within our framework to balance multi-layer saliency attribution (see Section III). This yields more interpretable visualizations by reducing the influence of extreme activations without discarding valuable multi-layer features.

Our proposed method, **Winsor-CAM**, builds on this insight by applying Winsorization to the importance

scores across all convolutional layers, combined with interpolation and weighted aggregation. Unlike existing approaches, Winsor-CAM introduces a *human-controllable parameter* that adjusts the percentile threshold, offering dynamic control over the balance between low- and high-level feature contributions. This allows users to explore explanations across semantic layers, supporting both expert diagnosis and interpretability in sensitive applications. In the next section, we formalize the Winsor-CAM framework, detailing how layer-wise Grad-CAM maps are aggregated and modulated via percentile-based Winsorization to produce multi-layer, tunable visual explanations.

III. Methodology

This section presents the formal definition of Winsor-CAM, our proposed method for producing interpretable, multi-layer saliency maps. We first provide a brief recap of the Grad-CAM algorithm, which forms the foundation of our approach. We then describe how Winsor-CAM generalizes Grad-CAM to aggregate feature relevance across all convolutional layers, introducing both improved stability through statistical Winsorization and user-level tunability via a percentile threshold.

A. Grad-CAM Revisited

Gradient-weighted Class Activation Mapping (Grad-CAM) [11] is a widely used technique for visualizing class-specific discriminative regions in CNNs. Grad-CAM computes a heatmap by computing the gradient of the target class score with respect to the feature maps in the final convolutional layer, and then uses these gradients to weight the importance of each feature map. Specifically, for a given class c, the importance weight for the k^{th} filter is defined as:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k},$$

where y^c is the score for class c, A_{ij}^k is the activation of feature map k at spatial location (i, j), and Z is the normalization factor which is equal to the total number of spatial locations in the map. These weights (α_k^c) are then linearly combined with the feature maps, followed by a ReLU activation:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right),$$

where $L^c \in \mathbb{R}^{H \times W}$ denotes the resulting class-specific localization map. The resulting heatmap ($L^c_{\text{Grad-CAM}}$) is typically upsampled to the input resolution and normalized for visualization. This pipeline for Grad-CAM is visualized in Fig. 3, with the corresponding pseudocode provided in Algorithm 1. As Grad-CAM generally focuses only on the final convolutional layer, it potentially misses important low- and mid-level features. Winsor-CAM overcomes this by extending Grad-CAM to all convolutional layers in the network.

B. Winsor-CAM: Layer-Wise Saliency Aggregation via Controlled Importance Scaling

Winsor-CAM generalizes Grad-CAM by aggregating class-specific saliency information from all convolutional layers in a CNN, rather than relying solely on the final layer. To ensure stable and interpretable visualizations, it incorporates statistical Winsorization to suppress outlier importance values and balance contributions across the network depth. As provided in Algorithm 1, our method proceeds through six stages: (*1*) layer-wise Grad-CAM computation, (*2*) spatial alignment via interpolation, (*3*) importance score extraction, (*4*) Winsorization-based outlier suppression, (*5*) normalized layer weighting, and (*6*) final saliency fusion.

Step 1. Grad-CAM Computation Per Layer: Let a CNN have n convolutional layers indexed by $i \in [1, n]$. For each layer i:

- $A^i \in \mathbb{R}^{C_i \times H_i \times W_i}$ are the feature maps,
- $A_k^i \in \mathbb{R}^{H_i \times W_i}$ is the k-th channel,
- C_i is the number of channels
- $y^c \in \mathbb{R}$ is the logit for class c.

The gradient of y^c w.r.t. A_k^i at spatial location $(u, v) \in [1, H_i] \times [1, W_i]$ is given by:

$$G^c_{i,k}(u, v) = \frac{\partial y^c}{\partial A_k^i(u, v)}. \tag{1}$$

To compute the importance of channel k at layer i for class c, we apply global average pooling over the gradient map:

$$\alpha^c_{i,k} = \frac{1}{Z_i} \sum_{u=1}^{H_i} \sum_{v=1}^{W_i} G^c_{i,k}(u, v), \quad \text{where } Z_i = H_i \cdot W_i. \tag{2}$$

The class-specific Grad-CAM map at layer i can be expressed as,

$$L^c_{\text{Grad-CAM},i}(u, v) = \text{ReLU}\left(\sum_{k=1}^{C_i} \alpha^c_{i,k} \cdot A^i_k(u, v)\right). \tag{3}$$

This heatmap reflects the spatial locations where features in A^i contribute positively to the prediction y^c. The ReLU ensures that only positively contributing regions are retained, following the original Grad-CAM formulation.

Step 2. Spatial Alignment via Interpolation: Since $L^c_{\text{Grad-CAM},i}$ from each layer i has spatial dimensions (H_i, W_i), we upsample all maps to a common resolution (H, W) to enable layer-wise aggregation:

$$\mathcal{W}^c_i = \text{Interp}(L^c_{\text{Grad-CAM},i}, H, W), \tag{4}$$

where $H = \max_i(H_i)$ and $W = \max_i(W_i)$. Interp$(\cdot)$ denotes a spatial interpolation function (e.g., bilinear or nearest-neighbor) applied to align all maps spatially. Let $\mathcal{W}^c = \{\mathcal{W}^c_i\}_{i=\{1,\ldots,n\}}$ denote the set of interpolated Grad-CAM maps for all layers.

Step 3. Per-Layer Importance Score Aggregation: To quantify the overall relevance of each layer i to class c, we

Algorithm 1 Winsor-CAM: Human-Tunable Visual Explanations from Deep Networks via Layer-Wise Winsorization

Require: Trained CNN model f, input image I, target class c, Winsorization percentile p, aggregation method $\in \{\text{mean}, \max\}$
Ensure: Winsor-CAM heatmap $\mathcal{W}^c_{\text{Winsor-CAM}}$

Initialize lists: $\mathcal{W}^c \leftarrow [\,]$, $\Gamma^c \leftarrow [\,]$
Let (H_i, W_i) be the spatial size of the i^{th} convolutional layer's output feature map
Let $(H, W) \leftarrow \max_{i=1}^{n}(H_i, W_i)$
for each convolutional layer $i = 1$ to n **do**
  Compute activations A^i // Step 1
  Compute gradients $G^c_{i,k} = \frac{\partial y^c}{\partial A^i_k}$
  $\alpha^c_{i,k} \leftarrow \frac{1}{H_i \cdot W_i} \sum_{u,v} G^c_{i,k}(u, v)$
  $L^c_i \leftarrow \text{ReLU}\left(\sum_k \alpha^c_{i,k} \cdot A^i_k\right)$
  $\mathcal{W}^c_i \leftarrow \text{Interp}(L^c_i, H, W)$ // Step 2
  $\Gamma^c_i \leftarrow \text{ReLU}(\text{mean/max}(\alpha^c_{i,k}))$ // Step 3
end for
Let $\Gamma^+ \leftarrow [\Gamma^c_i \mid \Gamma^c_i > 0]$ // Step 4
Let $T \leftarrow \text{Quantile}(\Gamma^+, p)$
for each layer $i = 1$ to n **do**
  $\Gamma^{c,\text{win}}_i \leftarrow \min(\Gamma^c_i, T)$ if $\Gamma^c_i > 0$, else 0
end for
Let $x_{\min} \leftarrow \min(\Gamma^+)$, $x_{\max} \leftarrow \max(\Gamma^+)$ // Step 5
for each layer $i = 1$ to n **do**
  $\tilde{\Gamma}^c_i \leftarrow 0.1 + \frac{\Gamma^{c,\text{win}}_i - x_{\min}}{x_{\max} - x_{\min}} \cdot 0.9$ if $\Gamma^{c,\text{win}}_i > 0$, else 0
end for
$\mathcal{W}^c_{\text{Winsor-CAM}} \leftarrow \sum_{i=1}^{n} \tilde{\Gamma}^c_i \cdot \mathcal{W}^c_i$ // Step 6
return $\mathcal{W}^c_{\text{Winsor-CAM}}$

compute a scalar Γ^c_i from its filter-wise weights $\alpha^c_{i,k}$. This can be defined using either of the following: mean or max aggregation. Let $\Gamma^c = \{\Gamma^c_1, \ldots, \Gamma^c_n\}$ be the vector of per-layer importance scores. These scores will guide how strongly each layer's heatmap contributes to the final visualization. Mean aggregation is defined as:

$$\Gamma^c_i = \text{ReLU}\left(\frac{1}{C_i}\sum_{k=1}^{C_i} \alpha^c_{i,k}\right). \tag{5}$$

Similarly, max aggregation can be expressed as,

$$\Gamma^c_i = \text{ReLU}\left(\max_{k \in \{1,\ldots,C_i\}} \alpha^c_{i,k}\right). \tag{6}$$

Importance values of zero for a layer do not indicate full irrelevance of that layer, but rather that features found within a layer do not tend to contribute to the class c in question as much as layers with higher importance scores. This is particularly useful in cases where a model may have learned to ignore certain features or patterns that are not relevant to the task at hand.

In practice, mean aggregation tends to yield lower importance scores for a larger number of layers, reflecting more conservative relevance estimates. In contrast, max aggregation often results in a greater number of layers receiving high importance scores, highlighting the most dominant filter in each layer. The choice between mean and max aggregation depends on the desired trade-off between interpretability and granularity in the resulting heatmap.

Step 4. Winsorization of Importance Scores: Classical Winsorization replaces values below the lower p^{th} and above the upper $(100 - p)^{\text{th}}$ percentiles with the respective threshold values. In contrast, we apply **one-sided upper clipping** at the p^{th} percentile, computed only over nonzero scores, to suppress dominant layers while preserving inactive ones. To prevent deeper layers with large Γ^c_i—a common occurrence in models such as those evaluated in this work, as shown in Figs. 2 and 4—from disproportionately influencing the final saliency map, we apply this process.

Let $\Gamma^+ = \{\Gamma^c_i \in \Gamma^c \mid \Gamma^c_i > 0\}$ be the set of nonzero importance scores as to allow for a threshold that is not impacted by layer importance values set to zero. We define the upper Winsorization threshold T as a user-chosen p^{th} percentile of Γ^+ (i.e., the set of positive importance scores). T is computed as follows:

$$T = \text{Quantile}(\Gamma^+, p). \tag{7}$$

We then, apply clipped thresholding using this found upper threshold T to each Γ^c_i:

$$\Gamma^c_{i,\text{winsor}} = \begin{cases} \min(\Gamma^c_i, T), & \Gamma^c_i > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{8}$$

Let $\Gamma^c_{\text{winsorized}} = \{\Gamma^c_{i,\text{winsor}}, \ldots, \Gamma^c_{n,\text{winsor}}\}$, or the set of all Winsorized importance values for every i^{th} layer for class c.

This step suppresses extreme values using a user-defined percentile p, controlling the influence of outliers while preserving zero values. The Winsorization process limits the effect of large layer-wise importance values, which might otherwise skew importance scores and produce misleading visualizations.

Step 5. Min-Max Normalization with Zero Preservation: To ensure interpretability and bounded contributions, we normalize $\Gamma^c_{\text{winsorized}}$ to a fixed range $[L, H]$ (default $[0.1, 1.0]$). This is done as follows:

$$\tilde{\Gamma}^c_i = \begin{cases} L + \frac{(\Gamma^c_{i,\text{winsor}} - x_{\min})}{x_{\max} - x_{\min}}(H - L), & \Gamma^c_{i,\text{winsor}} > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{9}$$

where $x_{\min} = \min \Gamma^+$ and $x_{\max} = \max \Gamma^+$. Let $\tilde{\Gamma}^c = \{\tilde{\Gamma}^c_i\}_{\{i=1,\ldots,n\}}$ denote the final set of per-layer importance weights after normalization.

By assigning zero to non-positive importance scores and mapping positive scores to the specified range (excluding zero), this step ensures that only layers with positive contributions influence the final heatmap. At the same time, it preserves the semantic meaning of zero-valued weights, indicating irrelevance. The parameters L and H can be tuned to balance visual contrast and interpretability. For example, the default choice $L = 0.1$ and $H = 1.0$ offers a clear separation between active and inactive layers, while allowing true zero values to persist.

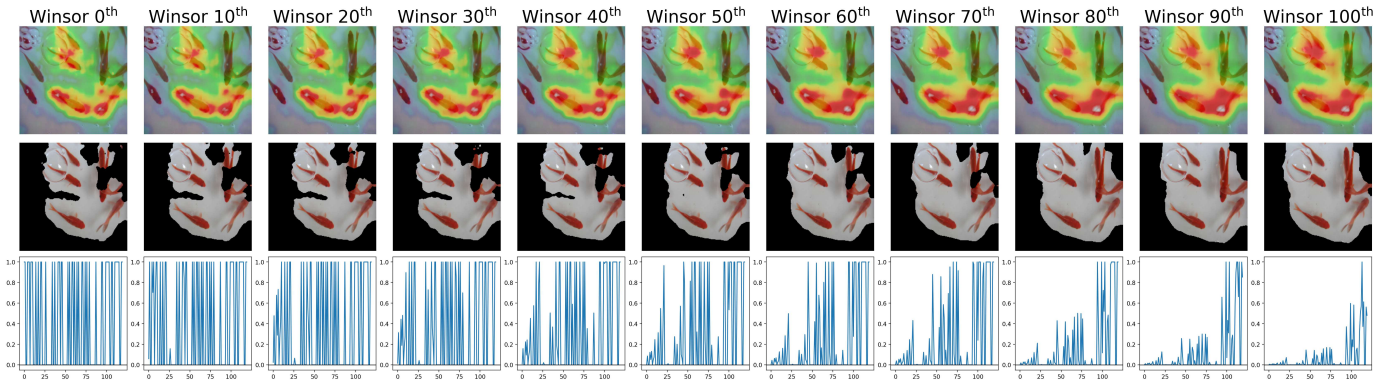

Fig. 4. Progression of Winsor-CAM visualizations on a sample image using DenseNet121, as the Winsorization percentile parameter p is varied from 0 to 100 in increments of 10. The top row shows the raw Winsor-CAM heatmaps, illustrating how saliency shifts from fine-grained details to broader object-level patterns as p increases. The middle row presents the corresponding binarized saliency masks after thresholding. The bottom row depicts the layer-wise importance distributions used to generate each heatmap; the x-axis corresponds to the layer index (ordered from early to late), and the y-axis indicates the relative importance assigned to each layer after Winsorization. Lower p-values suppress extreme scores and emphasize early-layer features (e.g., textures or edges), while higher p-values retain broader layer contributions, resulting in coarser, higher-level saliency. This figure demonstrates how Winsor-CAM enables semantic-level control over the granularity of visual explanations.

It is also worth noting that under maximum aggregation, a choice of $p = 0$ will typically result in $\Gamma_i^c = 1$ for all i. This occurs because $\max_{k\in\{1,\ldots,C_i\}} \alpha_{i,k}^c > 0$ generally holds for every layer i in a CNN. Consequently, the resulting heatmap becomes equivalent to the naïve mean of all Grad-CAM maps, as the importance scores are uniform across layers.

Step 6. Final Heatmap Construction: The final Winsor-CAM heatmap aggregates the interpolated Grad-CAM maps using the normalized importance scores by weighted linear combination of the interpolated Grad-CAM maps $\mathcal{W}_i^c$ using the Winsorized importance scores $\tilde{\Gamma}_i^c$:

$$\mathcal{W}^c_{\text{Winsor-CAM}}(u,v) = \sum_{i=1}^{n} \tilde{\Gamma}_i^c \cdot \mathcal{W}_i^c(u,v). \tag{10}$$

The result $\mathcal{W}^c_{\text{Winsor-CAM}} \in \mathbb{R}^{H\times W}$ reflects the spatially combined saliency from low- to high-level features across the entire network depth. Optionally, this map can be normalized to $[0,1]$ for better qualitative understanding, and overlaid as a heatmap on the original input for interpretability:

$$\text{Overlay}(I, \text{Normalize}(\mathcal{W}^c_{\text{Winsor-CAM}})). \tag{11}$$

The percentile threshold p, introduced in Step 4, provides a continuous control knob, enabling users to adjust how much weight is given to deep versus shallow features (with higher values of p typically emphasizing deeper layers and lower values emphasizing earlier ones). This pipeline is illustrated in Fig. 3 and the progression of Winsor-CAM with different p-values is shown in Fig. 4.

IV. Experiments

Evaluating explainability methods in deep learning remains inherently challenging, especially when the goal is to generate accurate visualizations of feature importance. Unlike conventional supervised learning metrics (e.g., accuracy or F1 score), visual explanation quality is inherently subjective and difficult to validate. As noted in [27], saliency-based methods may highlight features that align with internal model reasoning but diverge from human intuition. For instance, a CNN might associate wolves with forest-like backgrounds, leading the model to emphasize trees instead of the animal itself. Similarly, a digit classifier may fixate on specific geometric properties, such as the three points forming the digit "3," rather than its holistic shape. These examples highlight the gap between model attention and human-understandable concepts, underscoring the importance of both qualitative and quantitative evaluations in XAI. To address this, our evaluation incorporates both qualitative visualization analyses and quantitative performance metrics to assess the interpretability and spatial alignment properties of Winsor-CAM.

To evaluate Winsor-CAM, we conduct both qualitative and quantitative comparisons against standard Grad-CAM (final layer) and naïve mean aggregation of Grad-CAM maps from all convolutional layers. The backbone models used in our experiments include ResNet50, DenseNet121, VGG16, and InceptionV3 [28]–[31], each pretrained on the ImageNet dataset [32]. All inference experiments are conducted on the ImageNet validation set or the PASCAL VOC 2012 dataset [33], depending on the task.

A. Qualitative Evaluation

We first present a qualitative comparison of Winsor-CAM visualizations against standard Grad-CAM (final-layer only) and the naïve mean-layer aggregation baseline. As discussed in Step 4, the Winsorization step introduces a human-adjustable percentile parameter p to limit the dominance of extreme importance values

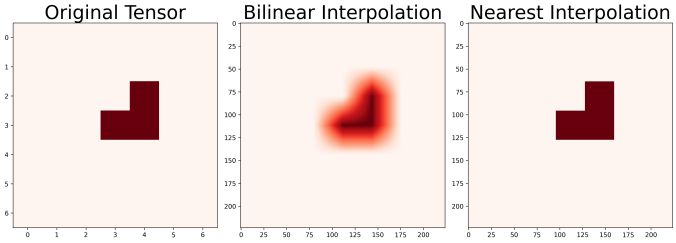

Fig. 5. Illustration of how interpolation affects the visual appearance of a feature map. Starting from a 7×7 resolution (left), the tensor is upsampled to 224×224 using nearest-neighbor interpolation (center) and bilinear interpolation (right). Nearest-neighbor preserves blocky structure, while bilinear produces smoother transitions.

across layers. This tunability allows users to control the semantic depth of explanations: lower p-values emphasize shallow features, whereas higher p-values prioritize deeper, abstract representations.

For qualitative results, we manually select representative values of p (where $p \in \{0, 100\}$) and examine the resulting heatmaps for alignment with semantically meaningful regions in the input image. In particular, we assess how effectively each method localizes class-discriminative features and suppresses background noise. First, we show how interpolation affects the visual appearance of saliency maps comparing how different layers of the same model look and analyzing how different sized feature maps look when upsampled to the same resolution. Finally, we show how Winsor-CAM can be used to visualize both high-level and low-level features in a way that is interpretable to a human.

1) Interpolation's Effect on Saliency Maps: A key factor that influences the visual appearance of saliency maps, regardless of the explanation method used, is the interpolation technique employed during upsampling. Since convolutional feature maps from intermediate layers are typically low-resolution (e.g., 7×7), they must be upsampled to match the input resolution (e.g., 224×224 for ImageNet images) before being visualized as heatmaps.

The choice of interpolation method can significantly affect the perceptual quality and interpretability of the resulting heatmaps. For example, using *nearest-neighbor interpolation* results in a coarse, blocky visualization resembling pixelation, which may obscure finer feature boundaries. In contrast, *bilinear interpolation* produces smoother transitions and more visually coherent maps, potentially aiding human interpretation. This phenomenon is illustrated in Fig. 5, which shows the impact of interpolation on a heatmap that could be produced by ResNet50.

Understanding the visual effect of interpolation is particularly important when comparing explanations qualitatively, as it can bias human perception of saliency granularity or precision. Therefore, consistent interpolation settings should be maintained across methods when evaluating interpretability.

As described in Eq. (4), Winsor-CAM upscales each

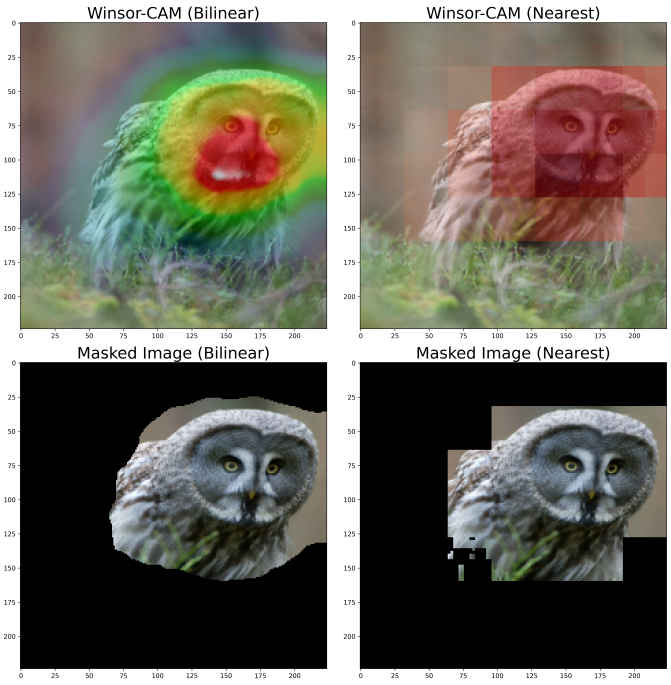

Fig. 6. Effect of interpolation method on Winsor-CAM outputs using DenseNet121. Left columns show results with bilinear interpolation; right columns use nearest-neighbor interpolation. Top row displays continuous heatmaps, while bottom row shows the corresponding binarized outputs. Differences highlight how interpolation choice can influence both visual appearance and localization structure.

intermediate Grad-CAM output to match the spatial resolution of the largest feature map in the network. These upsampled maps are then combined via a weighted linear sum, as defined in Eq. (10). The interpolation method used during this step can significantly affect both the visual quality of the resulting heatmaps and their behavior in subsequent post-processing steps, such as binarization. An example of the impact of interpolation on Winsor-CAM outputs is shown in Fig. 6.

Visualizations of Winsor-CAM and other methods that aggregate feature maps are typically less impacted by the choice of interpolation method than standard single-layer Grad-CAM, as they aggregate information from layers of different sizes. For instance, Grad-CAM heatmaps in ResNet-50 range from 112×112 to 7×7, corresponding to early and late convolutional layers, respectively. When all layers' feature maps are aggregated lower-level features from the earlier 112×112 tend to smooth the later higher-level 7×7 features. This is demonstrated in Fig. 1 where the final layer Grad-CAM is compared with multi-layer Winsor-CAM outputs using both bilinear and nearest-neighbor interpolation.

Nearest-neighbor interpolation preserves the original discrete structure of the feature maps and is arguably more faithful to the underlying convolutional architecture, which applies localized, grid-aligned filters. However, this results in blocky, pixelated outputs that may hinder human interpretation. In contrast, bilinear inter-

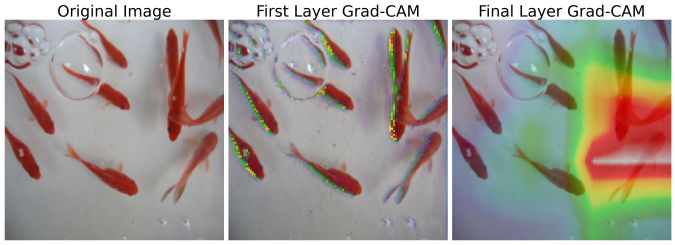

Fig. 7. Comparison of Grad-CAM outputs from the first and final convolutional layers of a DenseNet121 trained on ImageNet. The first layer highlights lower-level features such as edges, while the final layer captures higher-level, more spatially general regions.

polation produces smoother heatmaps with continuous transitions, which are generally more visually appealing and easier to interpret by humans.

This highlights an important trade-off between architectural faithfulness and human interpretability. While nearest-neighbor interpolation aligns more closely with the discrete nature of CNN operations, bilinear interpolation offers improved perceptual coherence. Most widely used Grad-CAM libraries, such as `pytorch-gradcam` [34], default to bilinear interpolation for visualization, reflecting this balance in practice.

2) Feature Hierarchy Visualization: To produce better visualizations, we seek to show both high-level and low-level features in a way that is interpretable to a human. As shown by [35], the earlier layers within a CNN capture low-level features (such as edges and colors), while later layers capture higher-level features (such as shapes and objects). This phenomenon is shown in Fig. 7, where the first layer Grad-CAM output of a DenseNet121 trained on Imagenet is shown alongside the last layer Grad-CAM output.

Winsor-CAM aims to capture both low- and high-level features relevant to a model's prediction by aggregating importance scores across all convolutional layers, using either mean or max strategies (Eqs. (5) and (6)). The user-controlled parameter p in Eq. (7) modulates this aggregation by attenuating extreme layer contributions: lower p-values emphasize early-layer, fine-grained lower-level features, while higher values prioritize deeper, more abstract higher representations. Fig. 4 illustrates this progression, with Winsor-CAM outputs shown for p-values from 0 to 100 in steps of 10. This tunable balance between shallow and deep features introduces a semantic trade-off that users can adjust based on task-specific interpretability needs.

To effectively visualize the output of Winsor-CAM across multiple p-values an animated GIF (Graphics Interchange Format) or video would best convey how the heatmap evolves. This dynamic representation would illustrate to a user how the model's focus shifts across the feature hierarchy as the p-value changes. This was not implemented in this work, as it is not a common practice in academic papers, and would be difficult to include in a static format. However, this is a possible future work that could be done to better visualize the output of Winsor-CAM. For instance, within the medical field one could create a user interface that allows a medical professional to manually adjust the p-value to show different features that the given model is focusing on as, perhaps, the final layer Grad-CAM output may not show all areas of medical significance.

B. Quantitative Evaluation

1) Evaluation Metrics: For quantitative evaluation, we adopt the Intersection over Union (IoU) metric to assess the localization performance of the generated saliency maps. IoU measures the overlap between the binarized saliency mask and the ground truth segmentation mask. The score is defined as:

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{12}$$

where TP, FP, and FN denote the true positives, false positives, and false negatives, respectively. TP corresponds to correctly identified foreground pixels, FP to incorrectly highlighted background pixels, and FN to missed ground truth regions.

Another metric considered in this work is the Euclidean distance between the Center-of-Mass (CoM) of the saliency map and the CoM of the ground truth mask. This metric complements IoU by measuring how closely the saliency map aligns with the object's true location. The CoM coordinates (x_c, y_c) for either a saliency heatmap or ground truth mask is defined as:

$$\begin{aligned} x_c &= \frac{\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} j \cdot I(i,j)}{\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} I(i,j)} \text{ and} \\ y_c &= \frac{\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} i \cdot I(i,j)}{\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} I(i,j)} \end{aligned}, \tag{13}$$

where $I(i,j)$ is the normalized intensity value at position (i,j), H is the height, and W is the width of the image. The normalization of I is performed as:

$$I(i,j) = \frac{M(i,j) - \min(M)}{\max(M) - \min(M) + \epsilon}, \tag{14}$$

where $M(i,j)$ represents the original intensity values derived from either a continuous saliency heatmap or a binary segmentation mask, and ϵ is a small constant (e.g., 10^{-6}) added to avoid division by zero. CoM for ground truth masks is computed using the binary segmentation masks, while the CoM for saliency maps is computed using the raw saliency maps before binarization. The Euclidean distance between the CoM of the saliency map and the ground truth mask is then computed using their respective (x_c, y_c) coordinates, and all units for Euclidean distance are given in pixels.

TABLE I
TRAINING AND VALIDATION METRICS FOR DIFFERENT MODELS

Model	Accuracy	F1	Precision	Recall
Validation				
ResNet50	0.8685	0.8651	0.8637	0.8685
DenseNet121	0.8416	0.8346	0.8361	0.8416
VGG16	0.8621	0.8582	0.8648	0.8621
InceptionV3	0.8567	0.8557	0.8599	0.8567
Train				
ResNet50	0.9979	0.9975	0.9980	0.9979
DenseNet121	0.9893	0.9872	0.9853	0.9893
VGG16	1.0000	1.0000	1.0000	1.0000
InceptionV3	0.9904	0.9884	0.9869	0.9904

2) Comparative Evaluation Protocol: For each test image, we compute the following aforementioned metrics:

- **Winsor-CAM IoU and CoM Distance:** The maximum IoU achieved and the minimum Euclidean distance between the CoM of the Winsor-CAM heatmap and the ground truth mask across all tested percentile values $p \in \{0 : 100 : 10\}$.
- **Grad-CAM IoU and CoM Distance:** The IoU and CoM distance computed from the standard Grad-CAM computed from the final convolutional layer.
- **Naïve Aggregation IoU and CoM Distance:** The IoU and CoM distance obtained by uniformly averaging Grad-CAM heatmaps across all convolutional layers.
- **All Other XAI Methods:** The IoU and CoM distance for other XAI methods such as Grad-CAM++ [12] and LayerCAM [13], and ShapleyCAM [18] are computed from the final convolutional layer.

These three configurations are compared to assess the efficacy of Winsor-CAM's tunable parameter p in adaptively optimizing localization performance on a per-image basis along with brief comparisons against other XAI methods. The objective is to demonstrate that Winsor-CAM not only surpasses fixed-output baselines in terms of IoU and localization accuracy but also introduces a flexible mechanism for incorporating human-guided interpretability.

Saliency maps are binarized using Otsu's thresholding method [36], which maximizes the between-class variance of pixel intensities to separate foreground from background at a global heatmap level. This binarization strategy was chosen because it simulates a human-tuned thresholding process without requiring manual calibration. In contrast, if one were to use a fixed threshold, such as 0.5, that would implicitly assume a sigmoidal relationship in the heatmap values, which may not hold in practice and would inadequately capture the variability in saliency map structures. The same binarization procedure is applied to baseline Grad-CAM outputs.

By selecting the optimal p for each image in the case of Winsor-CAM, we simulate a human-in-the-loop interpretability setting, where users can adjust the percentile to optimize interpretability. This is compared against fixed-output baselines which do not support parameter tuning (i.e. standard Grad-CAM and naïve mean aggregation).

TABLE II
COMPARISON OF WINSOR-CAM AND OTHER XAI METHODS ON CORRECTLY CLASSIFIED IMAGES (DENSENET121, MEAN AGGREGATION, BILINEAR INTERPOLATION)

Method	IoU	CoM Distance
Winsor-CAM	**0.469 ± 0.186**	**23.055 ± 16.379**
Grad-CAM	0.390 ± 0.173	25.684 ± 17.460
Grad-CAM++	0.368 ± 0.180	24.929 ± 15.609
LayerCAM	0.379 ± 0.174	24.331 ± 15.265
ShapleyCAM	0.377 ± 0.173	26.765 ± 17.528

We report the mean and standard deviation (image-to-image deviation) of IoU scores and CoM distance across the test dataset for each method. Additionally, we analyze how IoU and CoM distance vary as a function of p, providing insight into the trade-off between shallow and deep feature emphasis. These results demonstrate that Winsor-CAM consistently outperforms baseline methods in localization quality and offers flexibility for human-guided refinement.

3) Dataset and Training: The chosen dataset for this task is the PASCAL VOC 2012 dataset [33] as it provides ground truth maskings for all images. Since not all images in the PASCAL VOC 2012 dataset contain only a single class, we filter this dataset to only images that contain one class, as this ensures unambiguous correspondence between the saliency map and the target class. We are left with a dataset that has 933 images for training and 928 images for testing.

The classification head of each model was modified from 1000 to 20 output classes and a small amount of dropout was added to prevent overfitting. Each model was trained with the following configuration:

- **Optimizer:** AdamW
- **Loss Function:** Cross-Entropy
- **Learning Rate:** 0.0001 (with OneCycleLR scheduler)
- **Epochs:** 20
- **Batch Size:** 64
- **Input Size:** 224×224 (299×299 for InceptionV3)
- **Data Augmentations:**
  - Random horizontal flip ($p = 0.5$)
  - Random rotation (up to 7 degrees)
  - Color jitter (brightness = 0.2, contrast = 0.2, saturation = 0.2, hue = 0.1)
  - Random resized crop to 224×224 (scale = [0.8, 1.0], aspect ratio = [0.9, 1.1])
  - Random affine translation (up to 10% in both x and y directions)
  - Gaussian blur (kernel size = 3, $\sigma \in [0.1, 0.5]$)
- **Normalization:** Standard ImageNet mean and standard deviation
- **Hardware:** NVIDIA Tesla V100 GPU (32 GB)

The metrics for each tested model are shown in Table I.

4) Evaluation Methodology: Table II presents a comparison between Winsor-CAM and several established XAI methods—Grad-CAM++, LayerCAM, and ShapleyCAM—conducted under a single representative config-

TABLE III
PERFORMANCE METRICS FOR DIFFERENT INTERPOLATION METHODS (MEAN AGGREGATION) ON CORRECTLY CLASSIFIED IMAGES

Interp.	Model	IoU			Center-of-Mass Distance		
		Winsor-CAM	Final Layer	Avg Layer	Winsor-CAM	Final Layer	Avg Layer
Bilinear	ResNet50	**0.363 ± 0.145**	0.347 ± 0.178	0.319 ± 0.139	**21.998 ± 16.110***	25.293 ± 15.132	86.493 ± 20.664
	DenseNet121	**0.469 ± 0.186***	0.390 ± 0.173	0.428 ± 0.170	**23.055 ± 16.379**	25.353 ± 15.170	86.576 ± 20.764
	InceptionV3	**0.405 ± 0.195**	0.374 ± 0.196	0.370 ± 0.180	**27.667 ± 20.527**	34.538 ± 25.025	116.407 ± 27.287
	VGG16	**0.301 ± 0.149**	0.274 ± 0.169	0.272 ± 0.143	**24.868 ± 18.455**	33.531 ± 20.914	27.065 ± 18.505
Nearest	ResNet50	**0.372 ± 0.164**	0.347 ± 0.178	0.319 ± 0.139	**22.113 ± 16.229**	25.353 ± 15.170	86.576 ± 20.764
	DenseNet121	**0.456 ± 0.176***	0.379 ± 0.167	0.417 ± 0.161	**20.437 ± 15.593***	25.720 ± 17.510	86.449 ± 19.525
	InceptionV3	**0.399 ± 0.191**	0.369 ± 0.192	0.360 ± 0.173	**27.833 ± 20.697**	34.541 ± 25.065	116.550 ± 27.457
	VGG16	**0.290 ± 0.141**	0.265 ± 0.167	0.262 ± 0.136	**24.879 ± 18.463**	33.547 ± 20.946	27.056 ± 18.520

TABLE IV
PERFORMANCE METRICS FOR DIFFERENT INTERPOLATION METHODS (MAX AGGREGATION) ON CORRECTLY CLASSIFIED IMAGES

Interp.	Model	IoU			Center-of-Mass Distance		
		Winsor-CAM	Final Layer	Avg Layer	Winsor-CAM	Final Layer	Avg Layer
Bilinear	ResNet50	**0.377 ± 0.158**	0.354 ± 0.182	0.334 ± 0.150	**23.026 ± 16.278**	25.293 ± 15.132	86.493 ± 20.664
	DenseNet121	**0.453 ± 0.175***	0.390 ± 0.173	0.428 ± 0.170	**20.686 ± 14.940***	25.684 ± 17.460	86.384 ± 19.412
	InceptionV3	**0.396 ± 0.188**	0.374 ± 0.196	0.370 ± 0.180	**28.293 ± 21.142**	34.538 ± 25.025	116.407 ± 27.287
	VGG16	**0.331 ± 0.152**	0.274 ± 0.169	0.272 ± 0.143	**23.615 ± 17.378**	33.531 ± 20.914	27.065 ± 18.505
Nearest	ResNet50	**0.363 ± 0.145**	0.347 ± 0.178	0.319 ± 0.139	**23.055 ± 16.379**	25.353 ± 15.170	86.576 ± 20.764
	DenseNet121	**0.447 ± 0.167***	0.379 ± 0.167	0.417 ± 0.161	**20.740 ± 15.025***	25.720 ± 17.510	86.449 ± 19.525
	InceptionV3	**0.389 ± 0.183**	0.369 ± 0.192	0.360 ± 0.173	**28.540 ± 21.266**	34.541 ± 25.065	116.550 ± 27.457
	VGG16	**0.318 ± 0.146**	0.265 ± 0.167	0.262 ± 0.136	**23.620 ± 17.401**	33.547 ± 20.946	27.056 ± 18.520

TABLE V
PERFORMANCE METRICS FOR DIFFERENT INTERPOLATION METHODS (MEAN AGGREGATION) FOR INCORRECT PREDICTIONS

Interp.	Model	IoU			Center-of-Mass Distance		
		Winsor-CAM	Final Layer	Avg Layer	Winsor-CAM	Final Layer	Avg Layer
Bilinear	ResNet50	0.198 ± 0.164	**0.168 ± 0.148***	0.179 ± 0.148	38.318 ± 25.765	47.046 ± 29.815	**94.880 ± 30.920**
	DenseNet121	0.316 ± 0.206	0.390 ± 0.173	**0.277 ± 0.186**	31.475 ± 22.447	44.098 ± 32.174	**93.179 ± 30.067**
	InceptionV3	0.209 ± 0.164	0.158 ± 0.162	**0.177 ± 0.150**	47.247 ± 32.528	72.455 ± 48.009	**126.857 ± 41.691***
	VGG16	**0.170 ± 0.132**	0.211 ± 0.176	0.172 ± 0.128	39.085 ± 25.171	**41.269 ± 26.377**	38.597 ± 24.500
Nearest	ResNet50	0.190 ± 0.152	**0.167 ± 0.148**	0.172 ± 0.137	38.177 ± 25.810	47.151 ± 29.902	**94.969 ± 30.867**
	DenseNet121	0.306 ± 0.196	**0.225 ± 0.193**	0.269 ± 0.177	31.731 ± 22.589	44.200 ± 32.293	**93.223 ± 30.102**
	InceptionV3	0.201 ± 0.154	**0.156 ± 0.159***	0.172 ± 0.143	47.320 ± 32.459	72.636 ± 48.169	**126.792 ± 41.639***
	VGG16	**0.166 ± 0.127**	0.205 ± 0.168	0.168 ± 0.124	39.095 ± 25.168	**41.289 ± 26.374**	38.598 ± 24.512

TABLE VI
PERFORMANCE METRICS FOR DIFFERENT INTERPOLATION METHODS (MAX AGGREGATION) FOR INCORRECT PREDICTIONS

Interp.	Model	IoU			Center-of-Mass Distance		
		Winsor-CAM	Final Layer	Avg Layer	Winsor-CAM	Final Layer	Avg Layer
Bilinear	ResNet50	0.195 ± 0.150	**0.167 ± 0.148**	0.172 ± 0.137	38.310 ± 25.083	47.151 ± 29.902	**94.969 ± 30.867**
	DenseNet121	**0.230 ± 0.197**	0.232 ± 0.196	0.277 ± 0.186	32.530 ± 23.231	44.098 ± 32.174	**93.179 ± 30.067**
	InceptionV3	0.192 ± 0.157	**0.157 ± 0.162***	0.177 ± 0.150	51.095 ± 34.396	72.455 ± 48.009	**126.857 ± 41.691***
	VGG16	0.214 ± 0.157	**0.211 ± 0.176**	0.172 ± 0.128	35.944 ± 24.992	**41.269 ± 26.377**	38.597 ± 24.500
Nearest	ResNet50	0.194 ± 0.150	**0.167 ± 0.148**	0.172 ± 0.137	38.310 ± 25.083	47.151 ± 29.902	**94.969 ± 30.867**
	DenseNet121	0.295 ± 0.188	**0.225 ± 0.193**	0.269 ± 0.177	32.682 ± 23.230	44.200 ± 32.293	**93.223 ± 30.102**
	InceptionV3	0.188 ± 0.150	**0.156 ± 0.159***	0.172 ± 0.143	50.862 ± 34.290	72.636 ± 48.169	**126.792 ± 41.639***
	VGG16	0.208 ± 0.149	**0.205 ± 0.168**	0.168 ± 0.124	35.980 ± 24.989	**41.289 ± 26.374**	38.598 ± 24.512

uration (DenseNet121, mean aggregation, bilinear interpolation) due to space constraints. Beyond this broader comparison, we also evaluate Winsor-CAM against two internal Grad-CAM baselines: (*1*) the standard final-layer Grad-CAM output, and (2) a naïve mean aggregation of Grad-CAM maps across all convolutional layers. For these comparisons, inference was performed using p-values ranging from 0 to 100 in steps of 10. The corresponding results are shown in Tables III and IV for correctly classified images, and in Tables V and VI for misclassified ones. For each image, the Winsor-CAM columns report the highest IoU and lowest center-of-mass distance achieved across all p-values tested. In contrast, the Final Layer and Avg Layer columns reflect fixed outputs from the final-layer Grad-CAM and naïve mean aggregation, respectively. We also report the standard deviation across the test set for each metric to capture variability.

As all tested methods are inherently produced through the use of gradients and activations within a CNN, there

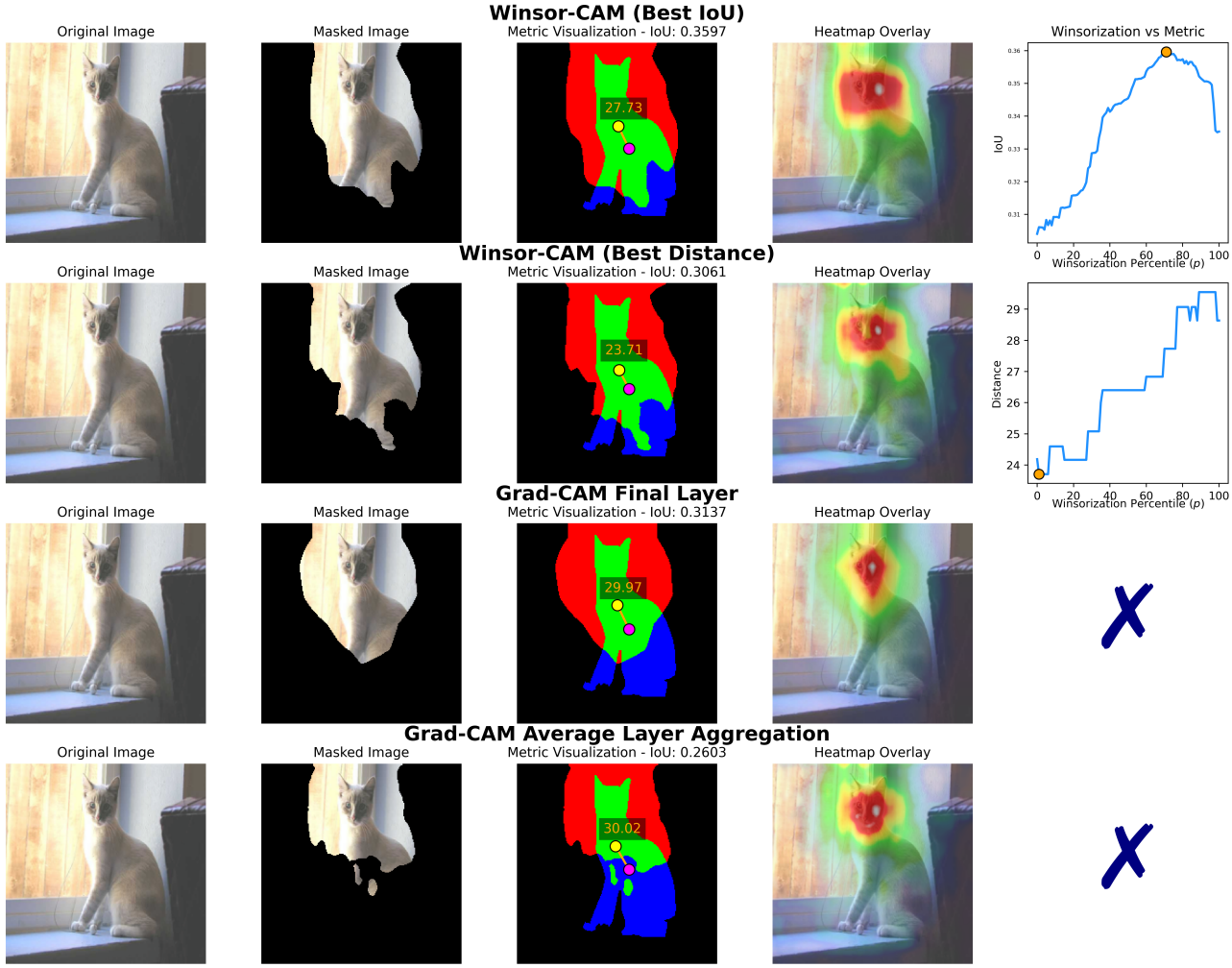

Fig. 8. Comparison of Winsor-CAM and Grad-CAM outputs on an image from the PASCAL VOC 2012 dataset, evaluated by IoU and Euclidean distance between ground truth and heatmap CoMs. The first two rows show Winsor-CAM results: (1) with the highest IoU and (2) with the lowest Euclidean distance. The last two rows display Grad-CAM outputs from the final layer and naïve mean aggregation across all layers, respectively. For each heatmap, the IoU visualization highlights true positives (green), false positives (red), false negatives (blue), and background (black). Euclidean distance is indicated by a magenta dot (ground truth center), a yellow dot (heatmap center), and an orange connecting line annotated with the distance value. The plots labeled "Winsorization vs. Metric" present IoU and Euclidean distance across all tested percentile (p) values for Winsor-CAM; these are not shown for Grad-CAM, which produces fixed outputs independent of p. Winsor-CAM results were generated using mean aggregation on a DenseNet121 trained on a PASCAL VOC 2012 subset as described in Section IV-B3.

may be variation in results between different runs on the same model. Consistent with prior findings [37], minor variability was observed in results when comparing outputs across different GPUs, and even across repeated runs on the same GPU. This variability is likely due to floating-point non-associativity and differences in execution order during parallel reduction operations, which are known to introduce nondeterminism in convolution-based deep learning systems during gradient computation. To mitigate this we attempt to set all random seeds and all attempt to set all flags for deterministic behavior in PyTorch. However, this does not guarantee that the results will be exactly the same across different runs. This deviation is difficult to quantify without systematically comparing runs across both identical and varied hardware/software configurations. This analysis was omitted as it falls outside the scope of this study as it would require a large amount of time to do properly. Furthermore, we run the same models on the same hardware and software configuration within the same general time period multiple times finding zero deviation in results across runs.

5) *Observations:* Table II shows that Winsor-CAM outperforms Grad-CAM++, LayerCAM, and ShapleyCAM under the tested configuration, further supporting its effectiveness in producing spatially aligned and semantically coherent explanations. In addition to this broader comparison, Tables III, IV, V, and VI show that Winsor-CAM consistently produces higher IoU values and lower Euclidean distances between the CoMs of the heatmap and the ground truth mask, compared to both the final-layer Grad-CAM output and the naïve mean aggregation of Grad-CAM outputs across layers. This holds across all

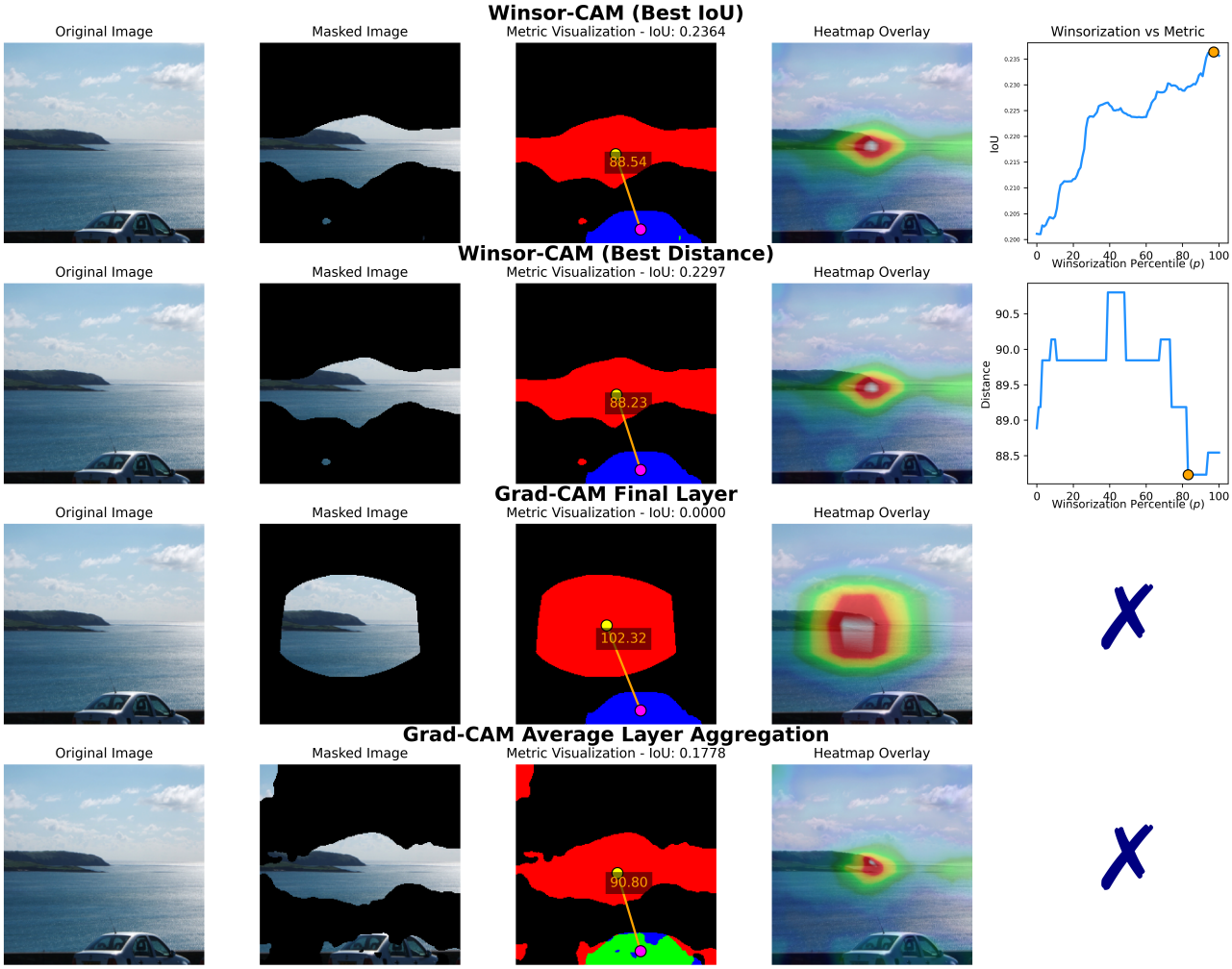

Fig. 9. Comparative analysis using the format of Fig. 8 for a misclassified image, where the model predicted "boat" instead of the ground truth "car." This example illustrates how the localization metrics and heatmaps differ in failure cases.

tested models and configurations.

We split results by prediction correctness because our primary interest lies in correctly classified cases. For incorrect predictions, IoU scores are generally lower, either because the model attends to features of the wrong class or fails to localize the target object entirely. A similar pattern is observed for the Euclidean distance between the CoMs of the ground truth segmentation mask and the predicted heatmap. Despite this, Winsor-CAM yields higher IoU and lower CoM distances than both final-layer Grad-CAM and naïve mean-layer aggregation, even on misclassified examples. This likely stems from the comparative evaluation protocol, defined in section IV-B2, selecting the optimal p-value for each case (i.e., maximizing IoU or minimizing distance), to enable consistent metric comparisons. Qualitative illustrations of these effects are shown in Figs. 8 and 9 for a correctly classified image and an incorrectly classified image, respectively. Each figure displays both IoU and CoM distance overlays, highlighting the improved spatial alignment achieved by Winsor-CAM.

A key finding is that the final-layer Grad-CAM tends to yield more accurate CoMs than naïve mean aggregation. However, it underperforms in terms of IoU compared to the mean aggregation of all layers. Winsor-CAM is able to produce both higher IoU values and lower Euclidean distances when compared to both the final-layer Grad-CAM and the naïve mean aggregation of every layer's Grad-CAM outputs. This shows that Winsor-CAM is able to better localize features that are indicative of the predicted class.

Notably, the per-image standard deviations of these metrics remain relatively high. This variability is partially attributable to differences in object size across the dataset, which interact with the scale and spatial resolution of feature maps at different convolutional layers. In particular, feature maps from deeper layers correspond to larger receptive fields. For small objects, these receptive fields often exceed the object's spatial extent, causing saliency maps to blur across multiple regions or emphasize irrelevant context. This scale mismatch between activation resolution and object size leads

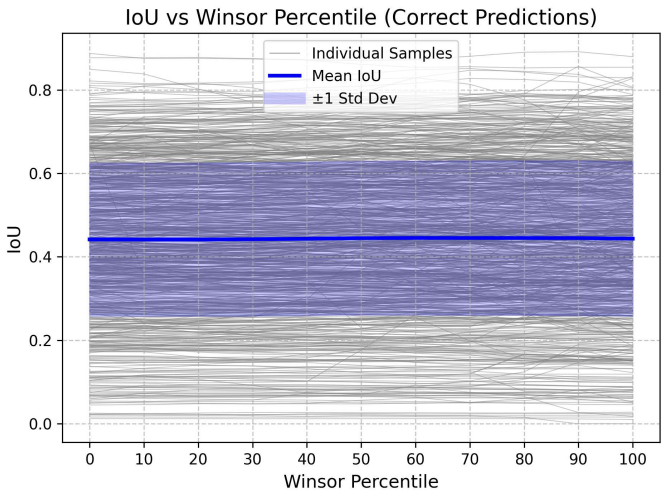

Fig. 10. Mean and standard deviation of IoU across all correctly classified images for DenseNet121 trained on ImageNet, shown for Winsor-CAM with p-values from 0 to 100 in increments of 10 using mean layer aggregation. Individual IoU values for each image and p-value are plotted in gray as to show large individual image deviations. Corresponding mean IoU values with standard deviations tabulated in Table VII.

to imprecise localization and contributes to performance variance in spatial alignment metrics. As [38] demonstrated, the effective receptive field in deep CNNs is significantly smaller and more centrally concentrated than its theoretical counterpart, resulting in late-layer activations that focus most strongly near the center of their input region. When the object of interest is small or located away from this central zone, especially in later convolutional layers with coarse resolution, the saliency output can become spatially diffuse or miss the object entirely. This phenomenon contributes to the high variance observed in spatial localization metrics and remains a significant challenge even for multi-layer saliency aggregation methods such as Winsor-CAM, which, while mitigating some effects, does not fully eliminate the impact of receptive field limitations.

The variability in optimal p-values across images, as shown in Fig. 10, also highlights the potential utility of Winsor-CAM in human-in-the-loop settings, where experts may wish to adjust the semantic depth of the explanation interactively. Although this tunability introduces variability, it enables adaptive explanations that can better align with user intent or task requirements. It is important to note that selecting the best p-value per image simulates an oracle-like setting. By reporting the best IoU or distance over a range of p-values, we illustrate the upper bound of Winsor-CAM's performance or, in other terms, its potential when a user or oracle selects the most effective p per image. Unlike standard Grad-CAM or fixed aggregation, which offer a single and static output, Winsor-CAM reveals a spectrum of possible explanations, adaptable to the interpretive needs of the user. These findings underscore the need for explanation methods that are not only quantitatively robust but also adaptable.

TABLE VII
Mean IoU and Mean Distance at Different Winsor-CAM Percentile Thresholds (p) for DenseNet121 (Correct Predictions).

p	IoU	CoM Distance
0	0.441 ± 0.182	22.778 ± 16.323
10	0.441 ± 0.183	22.758 ± 16.280
20	0.441 ± 0.183	22.721 ± 16.259
30	0.442 ± 0.183	22.673 ± 16.210
40	0.443 ± 0.183	22.565 ± 16.140
50	0.444 ± 0.184	22.475 ± 16.034
60	0.445 ± 0.184	22.360 ± 15.913
70	0.445 ± 0.185	22.252 ± 15.827
80	0.445 ± 0.185	22.132 ± 15.752
90	0.445 ± 0.185	21.993 ± 15.633
100	0.443 ± 0.185	21.993 ± 15.506

These results also point to the potential utility of Winsor-CAM in expert-in-the-loop settings, where users may wish to adapt explanations by selecting appropriate p-values per image. While not guided by a true oracle, the use of per-image p-value selection and Otsu thresholding serves to illustrate Winsor-CAM's capacity for generating tailored explanations in a domain-adaptive manner.

V. Conclusion and Future Work

This paper introduced Winsor-CAM, a human-tunable and novel extension of Grad-CAM that aggregates saliency across all convolutional layers using percentile-based Winsorization to suppress outliers and balance contributions. By introducing a user-adjustable parameter p, Winsor-CAM enables semantically controllable visual explanations and produces coherent heatmaps across the entire convolutional stack. Experiments across multiple CNN architectures demonstrate that Winsor-CAM consistently outperforms both Grad-CAM baselines and other CAM-based methods—including Grad-CAM++, LayerCAM, and ShapleyCAM—in terms of localization fidelity (IoU and center-of-mass accuracy). These findings highlight Winsor-CAM as an effective post-hoc interpretability tool for expert-in-the-loop applications, bridging the gap between automated attribution and human-controllable semantic tuning, and contributing to more trustworthy and transparent deep learning. Future work will explore adaptive selection of the Winsorization parameter, integration into interactive diagnostic interfaces, and evaluations on more complex and domain-specific datasets.

References

[1] S. U. Hamida, M. J. M. Chowdhury, N. R. Chakraborty, K. Biswas, and S. K. Sami, "Exploring the landscape of explainable artificial intelligence (xai): A systematic review of techniques and applications," *Big Data and Cognitive Computing*, vol. 8, no. 11, p. 149, 2024.

[2] L. Longo, M. Brcic, F. Cabitza, J. Choi, R. Confalonieri, J. Del Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger *et al.*, "Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions," *Information Fusion*, vol. 106, p. 102301, 2024.

[3] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[4] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision* (*ECCV*). Springer, 2014, pp. 818–833.

[5] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *Proceedings of the International Conference on Machine Learning* (*ICML*), 2019.

[6] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, "Generating visual explanations," in *Proceedings of the European Conference on Computer Vision* (*ECCV*), 2016.

[7] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata, "Grounding visual explanations," in *Proceedings of the European Conference on Computer Vision* (*ECCV*), 2018.

[8] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning* (*ICML*), vol. 70. PMLR, 2017, pp. 3145–3153.

[9] J. Wagner, J. M. Kohler, T. Gindele, L. Hetzel, J. T. Wiedemer, and S. Behnke, "Interpretable and fine-grained visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2019.

[10] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2016, pp. 2921–2929.

[11] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision* (*ICCV*), 2017, pp. 618–626.

[12] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision* (*WACV*). IEEE, 2018, pp. 839–847.

[13] P.-T. Jiang, C. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "Layercam: Exploring hierarchical class activation maps for localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 5875–5888, 2021.

[14] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li, "Axiom-based grad-cam: Towards accurate visualization and explanation of cnns," 2020. [Online]. Available: https://arxiv.org/abs/2008.02312

[15] T. Yamauchi, "Spatial sensitive grad-cam++: Improved visual explanation for object detectors via weighted combination of gradient map," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8164–8168.

[16] H.-C. Dong, Y. Jiang, Y. Huang, J. Liao, B. Liu, D. Ye, and G. Liu, "Rethinking class activation maps for segmentation: Revealing semantic information in shallow layers by reducing noise," *arXiv preprint arXiv:2308.02118*, 2023.

[17] T. Yamauchi, H. Kera, and K. Kawamoto, "Explaining object detectors via collective contribution of pixels," *arXiv preprint arXiv:2412.00666*, 2024.

[18] H. Cai, "Cams as shapley value-based explainers," *arXiv preprint arXiv:2501.06261*, 2025.

[19] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," 2017. [Online]. Available: https://arxiv.org/abs/1706.03825

[20] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam, "Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models," 2019. [Online]. Available: https://arxiv.org/abs/1908.01224

[21] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning* (*ICML*), vol. 70. PMLR, 2017, pp. 3319–3328.

[22] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview," *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 193–209, 2019.

[23] V. Buono, P. S. Mashhadi, M. Rahat, P. Tiwari, and S. Byttner, "Expected grad-cam: Towards gradient faithfulness," *arXiv preprint arXiv:2406.01274*, 2024.

[24] M. T. Ribeiro, S. Singh, and C. 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* (*KDD*), 2016, pp. 1135–1144.

[25] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems* (*NeurIPS*), vol. 30, 2017, pp. 4765–4774.

[26] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," in *British Machine Vision Conference* (*BMVC*), 2018.

[27] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2660–2673, 2017.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2016, pp. 770–778.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2017, pp. 2261–2269.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2016, pp. 2818–2826.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec 2015. [Online]. Available: https://doi.org/10.1007/s11263-015-0816-y

[33] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[34] J. Gildenblat, "pytorch-grad-cam: Gradient-based class activation maps in pytorch," https://github.com/jacobgil/pytorch-grad-cam, 2021, accessed: 2025-04-29.

[35] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015. [Online]. Available: https://arxiv.org/abs/1506.06579

[36] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[37] S. Shanmugavelu, M. Taillefumier, C. Culver, O. Hernandez, M. Coletti, and A. Sedova, "Impacts of floating-point non-associativity on reproducibility for hpc and deep learning applications," 2024. [Online]. Available: https://arxiv.org/abs/2408.05148

[38] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 4905–4913.