SemScore: Practical Explainable AI through Quantitative Methods to Measure Semantic Spuriosity

Jovin Leong
Home Team Science & Technology Agency
jovin_leong@htx.gov.sg

Chen Wei May
Home Team Science & Technology Agency
chen_wei_may@htx.gov.sg

Tan Tiong Kai
Home Team Science & Technology Agency
tan_tiong_kai@htx.gov.sg

Abstract

Mispredictions caused by spuriosity and flawed model reasoning remain challenges in predictive machine learning and artificial intelligence; Explainable AI (XAI) aims to mitigate these issues by tackling model interpretability and explanability, guided by principles such as explanation accuracy and knowledge limits. However, these principles are largely qualitative, leaving researchers with few actionable tools to quantify issues like spuriosity, limiting their usefulness in AI development and research. This gap is problematic as it leaves researchers to perform laborious, manual techniques to assess individual model predictions—assessments that are subject to errors of human judgment. We introduce SemScore, an extensible toolkit that applies a novel method to determine the semantic relevance of models by quantifying visual explanation methods through semantic segmentation datasets. By comparing visual explanation methods against ground-truth semantics, SemScore evaluates models on spuriosity, enabling researchers to systematically measure and quantify the semantic understanding of models. This provides a useful and actionable toolkit for understanding model biases and behavior. We apply SemScore to various computer vision domains and demonstrate that SemScore can effectively evaluate and discern between models based on their semantic reasoning capabilities. As the first practical method for quantifying semantic understanding through spuriosity analysis, SemScore significantly advances the capabilities for XAI research. We release the SemScore toolkit and experimentation code publicly to provide researchers with the means to build more semantically relevant models and to extend our work into additional domains.[1]

1 Introduction

The increasing depth and complexity of modern machine learning models have unlocked powerful capabilities at the cost of interpretability [30, 44]. While improving prediction accuracy is a common focus, a more subtle yet critical issue is often ignored: correct model predictions made for the wrong reasons. These are cases in which models produce correct predictions based on irrelevant or spurious features, suggesting a fortunate coincidence at best and a fundamental misunderstanding of the predictive task at worst. These errors in reasoning, driven by spurious or irrelevant features, can mask poor generalization and present a false sense of reliability, especially under domain shifts [46].

[1] https://github.com/JovinLeong/SemScore

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

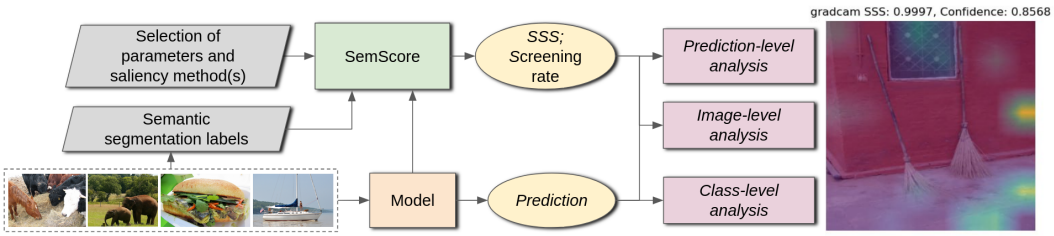

Figure 1: **Overview of the SemScore workflow**. Given a model, saliency method, and semantic segmentation labels, SemScore computes the *SSS* by comparing saliency map activations with semantically relevant regions. This enables multi-level analysis at the prediction, image, and class levels. Example: PVT-v2 [56] correctly predicted *broom* class label; the high *SSS* (0.9997) computed using GradCAM indicates a strong reliance on semantically irrelevant features despite high confidence (0.8568), signaling spurious reasoning.

The essentiality of understanding model reasoning in high-stakes domains like healthcare, autonomous systems, and finance, understanding model reasoning [6] has driven interest in Explainable AI (XAI), yet most current methods rely on qualitative visualizations that are subjective, biased, and labor-intensive. As a result, they fall short of delivering the scalable, objective insights necessary to build trustworthy AI systems. Thus, there remains a critical need for generalizable, quantitative metrics that can objectively assess and compare model reasoning at scale.

To address these challenges, we introduce SemScore, a toolkit for quantifying model semantic spuriosity, which can be understood as the extent to which models rely on semantically irrelevant concepts when making predictions. We define Semantic Spuriosity as the degree to which a model relies on features that are not part of semantically relevant regions associated with the predicted class. The semantically relevant region is specified by the semantic segmentation mask, which represents the ground truth label for a given region. SemScore is a toolkit that supports relative model comparisons using Semantic Spuriosity Score (*SSS*); *SSS* measures how much a model's attention lies outside relevant regions, which provides an effective, quantitative means to assess whether models are making predictions for the right reasons. Our contributions are threefold:

- **Semantic Spuriosity Score (*SSS*)**: A novel metric to quantify semantic understanding in models at multiple levels: prediction, image, and class.
- **SemScore toolkit**: an open-source framework that enables researchers to readily use our metrics to evaluate the semantic performance of their vision models.[2]
- We apply SemScore across CNNs, Vision Transformers, and Vision-Language Models to uncover meaningful differences in semantic reasoning across architectures and tasks.

SemScore moves XAI beyond subjective interpretation, offering a rigorous, actionable path to transparent and trustworthy AI. We further show that *SSS* captures information orthogonal to prediction confidence, offering insight into model reasoning that is not reflected in output scores alone.

2 Related work

2.1 Saliency-Based Explainability

Saliency methods are central to visual explainability, offering post-hoc insights into model behavior. Gradient-based techniques such as Grad-CAM [44], Integrated Gradients [48], and their variants (e.g., LayerCAM, Grad-CAM++) aim to highlight input regions important for a model's decision. Gradient-free approaches such as ScoreCAM [54], Eigen-CAM [35], and AblationCAM [9] provide alternatives without relying on model internals, bypassing gradients to improve compatibility across architectures. These methods have been widely adopted in XAI due to their model flexibility and visual interpretability. Yet, growing evidence has called into question the faithfulness and reliability of saliency maps. Notably, Adebayo et al. [2], Kim et al. [23], and Zhou et al. [60] found that many saliency methods are sensitive to randomization and may fail to capture true model reasoning.

[2] `https://github.com/JovinLeong/SemScore`

These critiques highlight that saliency outputs can be misleading when interpreted at face value (e.g., via the "pointing game"). Rather than validate saliency maps themselves, SemScore addresses a complementary question: *given the saliency of a prediction, how well does this saliency align with task-relevant semantics?* This shifts the focus from attribution correctness to semantic congruence, allowing researchers to compare models' reasoning regardless of saliency reliability. SemScore's extensible design accommodates various attribution methods and model types precisely to account for method-model compatibility.

2.2 Detecting and Quantifying Spuriosity

A related research direction focuses on identifying and mitigating spurious correlations or shortcut behaviours in deep models. This includes techniques for dataset debiasing, causal inference, and feature attribution analysis. For example, SpRAy [24] applies spectral clustering to Layer-wise Relevance Propagation to reveal "Clever Hans" predictors—models that learn to rely on irrelevant visual artifacts that co-occur with certain labels. Other recent efforts, such as Friedrich et al. [12], introduce taxonomies and metrics to detect and mitigate shortcut learning. Steinmann et al. [46] provide a comprehensive taxonomy of techniques for detecting, analyzing, and mitigating model shortcuts. Though primarily aimed at intervention and mitigation, these approaches also rely on identifying model reliance on unintended cues. SemScore complements this line of work by offering a scalable, model-agnostic diagnostic quantitative tool for post-hoc detection of semantic alignment. It enables analysis of model reasoning without retraining or manual dataset curation, supporting fine-grained inspection across predictions and classes.

Spuriosity in XAI has received less attention. Moayeri et al. [34] proposed human-guided spuriosity rankings to assess model biases caused by reliance on spurious cues in ImageNet [42]. Le et al. [25] introduced the neuron spurious score to quantify a neuron's dependence on spurious features to study shortcut learning, though limited by dependence on Grad-CAM. Unlike these works, which aim to detect or correct spuriosity at the model level, SemScore provides a post-hoc, model-agnostic evaluation of semantic alignment using saliency maps and semantic segmentation labels.

2.3 Explainability in Transformer and Vision-Language Architectures

For transformer-based models, attention-based methods aim to interpret how information flows across layers. Attention Rollout and Attention Flow [1] model information flow in the network as a directed acyclic graph, while Gradient Attention Rollout (GAR) by [14] extends Attention Rollout by incorporating gradients to refine token relevance. Dynamic Accumulated Attention Map (DAAM) [27] accumulates spatial features across layers and combines them with channel-wise importance to generate fine-grained attention maps. These methods offer insight into hierarchical reasoning but remain limited in evaluating semantic validity. For vision-language models (VLMs), explainability remains an emerging area. Works such as Hashmi et al. [16] evaluate VLMs within the specialized domain of medical imaging, while Suh et al. [47] explore the efficacy of conventional saliency methods in interpreting VLMs. Others assess semantic continuity—the principle that similar inputs should yield similar explanations [19]. However, most approaches stop at visual inspection, lacking metrics to quantify semantic misalignment or domain overfitting. SemScore addresses this by introducing a unified framework to measure semantic spuriosity across VLMs and unimodal models, revealing whether model focus aligns with annotated semantically relevant regions.

2.4 Semantic Explanations and Segmentation-Based Approaches

In XAI, semantics refer to explanations grounded in human-understandable concepts. Semantic segmentation-based attribution extends saliency analysis by aligning model attention with object or region-level labels. For example, Seg-Grad-CAM [53] and Seg-XRes-CAM [15] adapt Grad-CAM to pixel-wise outputs for semantic tasks. Other approaches D-RISE [37] and D-CLOSE [52] offer black-box explanations for object detectors using randomized masking, but are sensitive to noise, hyperparameters, and scene complexity. SemScore unifies and extends these ideas, offering a standardized, quantitative measure of semantic alignment applicable to any task with segmentation annotations. It supports comparative benchmarking across saliency methods and model types, providing actionable insight into model reasoning. Despite the proliferation of attribution methods, key gaps remain in how explanations are evaluated. Saliency outputs, when visually assessed, limit

reproducibility and scalability [50, 36]. While toolkits like Quantus [45] provide standardized metrics (e.g., faithfulness, sensitivity), they do not assess whether explanations align with task-relevant semantics. Similarly, spatial metrics (e.g., explanation accuracy) capture overlap but not conceptual alignment. SemScore addresses this by introducing *SSS*, a quantitative metric to evaluate how well model explanations align with semantic regions. It supports multi-level aggregation (per-prediction, per-class, per-image) to enable scalable, model-agnostic analysis of reasoning quality.

3 The SemScore Toolkit

The SemScore toolkit enables scalable, quantitative analysis of semantic spuriosity in vision models, including CNNs, transformers, and VLMs. It supports any model with convolution or attention layers by comparing model-generated saliency maps with semantic segmentation masks. This comparison yields the Semantic Spuriosity Score (*SSS*), a metric that quantifies the extent to which a model's prediction relies on semantically irrelevant regions. *SSS* captures how well a model's focus aligns with human-understandable concepts. This involves comparing saliency maps to segmentation masks (Figure 2). Using semantic segmentation datasets, SemScore enables systematic evaluation of a model's semantic reasoning across multiple vision tasks, reducing reliance on subjective visual inspection of saliency maps. The toolkit provides modular components to configure datasets, models, saliency methods, and evaluation parameters. Once a configuration is selected, SemScore computes *SSS* and provides multi-level aggregations at the prediction, image, and class levels.

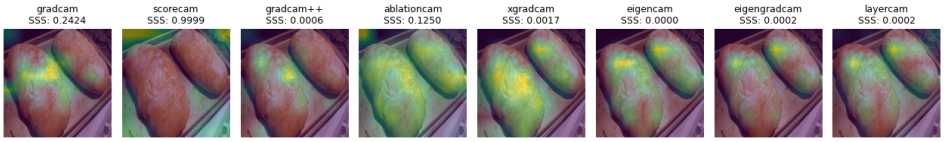

Figure 2: Variability of saliency maps and *SSS* across attribution methods. ScoreCAM fails to align with annotated regions and has a high *SSS* of 0.9999; LayerCAM and EigenCAM exhibit accurate activations with low *SSS*, indicating low spuriosity and strong semantic alignment.

Importantly, SemScore is designed for relative, not absolute evaluation. For any given model-dataset pair, suitable thresholding and saliency configurations need to be initially determined. These settings should then be held fixed when comparing model variants to ensure consistent and reproducible comparisons. This design mitigates the risk of researchers drawing contradictory conclusions under different parameterizations.

The utility of *SSS* inevitably depends on the quality of the initial configuration—a reflection of the semantic gap between human interpretation grounded in semantic labels and the model's internal reasoning as approximated by saliency maps. Because different architectures are best evaluated with different saliency methods, manual selection remains important to ensure compatibility with the model under analysis. SemScore eases this selection process through a simple, modular configuration process. All code, configurations, and experiment scripts are publicly released to support replication, adoption, and extension of the framework.[3]

3.1 Semantic Spuriosity Score (*SSS*)

The Semantic Spuriosity Score (*SSS*) is defined as the normalized difference between the total saliency mass within the semantically relevant region, defined as the combined semantic segmentation masks across all specified classes, and the saliency mass outside the semantically relevant region, normalized by the total saliency mass. *SSS* estimates the degree to which a model relies on irrelevant semantic features, allowing researchers to identify and measure spuriosity.

Let M be the saliency map, and let $R \subset \Omega$ denote the semantically significant region within the image domain Ω. We have an initial result SSS' in (1) where $SSS' \in [-1, 1]$. Thereafter, we linearly map SSS' to the range $[0, 1]$ and invert it by subtracting SSS' from 1 to obtain SSS where

[3] `https://github.com/JovinLeong/SemScore`

SSS $\in [0, 1]$ in (2):

$$SSS' = \frac{\sum_{x \in R} M(x) - \sum_{x \in \Omega \setminus R} M(x)}{\sum_{x \in \Omega} M(x)} \qquad SSS = 1 - \frac{SSS' + 1}{2} \tag{1,2}$$

The *SSS* measures the degree to which features deemed important by a model overlap with semantically irrelevant regions—areas unrelated to the target class. *SSS* aims to quantify the negative influence of irrelevant features on the model's prediction. A high score would mean the model may be relying on spurious cues or background artifacts stemming from incorrect or misleading attention.

SSS differs fundamentally from standard segmentation evaluation metrics such as IoU or pixel-wise accuracy. While segmentation metrics compare a predicted mask to the ground-truth mask to assess output correctness, *SSS* compares the saliency map of the model (a post-hoc explanation of where the model is focusing attention) to the ground-truth semantic segmentation mask to estimate semantic alignment of a model's reasoning to human labels. This is distinct from standard segmentation evaluation metrics like IoU or pixel-wise accuracy, which assess the degree of prediction correctness. Thus, *SSS* provides insight into model behavior and interpretability, even when predictions are accurate.

SSS is based on the premise that correct model reasoning aligns with task-relevant semantics. By measuring the overlap between saliency maps and ground-truth segments, *SSS* quantifies this alignment. High *SSS* implies attention to irrelevant areas, reflecting poor task localization or shortcut learning. Hence, *SSS* serves as a proxy for faithfulness, assuming saliency maps reflect true model reasoning.

3.2 Pixel Importance Thresholding and Aggregations

Saliency maps often contain noisy activations that obscure the model's true focus. To improve the signal-to-noise ratio, we apply thresholding techniques to filter out low-importance pixels. This process sharpens saliency maps by suppressing diffuse, irrelevant activations and preserving more meaningful ones. By doing so, the resulting *SSS* more accurately reflects a model's reliance on semantically relevant regions. Thresholding can be configured in either soft or hard modes, depending on whether original saliency values are preserved or binarized. The detailed mathematical formulation and thresholding function are provided in Appendix C.

Additionally, SemScore supports per-prediction, per-image, and per-class aggregation for multi-level insights into model reasoning. Per-prediction aggregation enables fine-grained analysis of individual outputs. Per-image aggregation summarizes reasoning quality across predictions in a single context. Per-class aggregation highlights class-specific spuriosity patterns to identify bias or underperforming classes. Together, these views offer a comprehensive framework for evaluating semantic alignment across tasks.

3.3 Significance of SemScore and *SSS*

SemScore provides a systematic, architecture-agnostic framework for evaluating semantic reasoning in vision models. By comparing saliency maps against semantic segmentation masks, it reveals subtle failure cases, such as correct predictions made for the wrong reasons, that standard metrics overlook. This makes SemScore especially valuable for AI safety, robustness, and bias analysis. *SSS* extends prior metrics like the "pointing game" by: (1) handling multi-class, multi-region inputs, (2) supporting soft thresholding for continuous saliency values, and (3) enabling flexible aggregation at the prediction, image, and class levels across complex tasks such as detection and vision-language grounding. Several cases illustrating the utility of SemScore are discussed in Section 4.5 with further cases discussed in Appendix A.

4 Experiments

All experiments were conducted on Amazon Web Services' g4dn.4xlarge instances with NVIDIA T4 GPUs and Deep Learning OSS Nvidia Driver Amazon Linux 2 Amazon Machine Image for PyTorch 1.13.1.[4]. We evaluate a range of publicly available computer vision models across different tasks, using default model hyperparameters. In each case, we compute *SSS* to quantify the extent to which a

[4] `https://aws.amazon.com/ec2/instance-types/g4/`

model relies on semantically irrelevant regions, where higher *SSS* values indicate greater spuriosity and poorer semantic alignment. We compute *SSS* at the per-prediction, per-image, and per-class score aggregation, but report only per-image results for brevity; full results are provided in Appendix D.

4.1 Image Classification

We evaluate leading image classification models trained on ImageNet-1K [42], using ImageNet-S [13], an extension of ImageNet-1K with semantic segmentation annotations for *SSS* computation. All validation images from ImageNet-S were passed through each model, and saliency maps were generated using various CAM methods.

Table 1: Per-image *SSS* for image classifiers on ImageNet-S [13]

Model	GradCAM ↓	Grad CAM++ ↓	Score CAM ↓	Layer CAM ↓	Eigen CAM ↓	Eigen GradCAM ↓	Ablation CAM ↓
DeiT Tiny [51]	0.1844	0.4423	0.3323	0.2402	0.6205	0.6174	0.2394
DeiT Small [51]	0.1759	0.4118	0.4418	0.1951	0.4140	0.4581	0.1828
DeiT Base [51]	0.3614	**0.3971**	0.5193	0.1724	0.5777	0.6472	0.3136
ViT Base [38]	0.2759	0.556	0.4272	0.2670	0.3813	0.8639	0.3864
FlexiViT Base [3]	**0.1436**	0.5802	**0.3019**	0.2159	0.3434	0.2691	0.2248
ConViT Small [8]	0.1650	0.4858	0.4939	**0.1379**	0.4778	0.2948	**0.1600**
Hiera-Base-Plus [43]	0.8035	0.8339	0.5303	0.5338	0.4655	0.3100	0.2789
PVT-v2 [56]	0.6203	0.6860	0.4339	0.2696	**0.2786**	**0.1991**	0.8272
Swin-Base [32]	0.3881	0.7480	0.5192	0.7967	0.3267	0.2286	0.8918

From Table 1, we observe substantial variation in spuriosity across both models and attribution methods. Notably, FlexiViT and ConViT-Small consistently yield the lowest *SSS* across most attribution methods despite their smaller model size, suggesting that architectural design, rather than model size, plays a more significant role in achieving semantic alignment. FlexiViT's content-adaptive token aggregation and ConViT's soft convolutional biases likely promote attention to task-relevant regions, improving robustness against spurious correlations. In contrast, Swin-Base and Hiera-Base-Plus exhibit the highest semantic spuriosity. These hierarchical vision transformers rely on window-based attention mechanisms that process local patches and aggregate information progressively. While efficient, this localized structure may limit global contextual understanding and lead to overfitting on background textures or co-occurring artifacts. Our results reveal clear disparities in semantic reasoning among ViT-based models; this is an indication that *SSS* provides insight into model behavior beyond accuracy, especially in safety-critical or generalization-sensitive applications.

4.2 Object Detection

We select leading object detection models trained on the COCO dataset [28] and evaluate them on the COCO-Stuff 10K validation dataset [4, 22]. All models achieved a detection coverage of 0.999, defined as the percentage of positive samples where the model outputs at least one detection.

Table 2: Per-image *SSS* results for object detectors on COCO-Stuff 10K [4, 22]

Model	GradCAM ↓	Grad CAM++ ↓	Layer CAM ↓	Eigen CAM ↓	Eigen GradCAM ↓	Ablation CAM ↓
FCOS [49]	0.7191	0.7003	0.6222	0.8094	0.6502	0.7399
SSDLite320 Large [18]	0.6873	0.7358	0.5882	0.6608	0.5832	0.7164
RetinaNet [29]	0.5753	0.5387	0.4079	0.6867	0.4236	0.5861
Faster R-CNN [40]	0.6583	0.6228	0.5122	0.7511	0.5167	0.6672
SSD300 [31]	**0.4955**	**0.3169**	**0.2580**	**0.5091**	**0.3444**	**0.4549**

The results from Table 2 show that SSD300 has consistently lower *SSS* than all other detectors evaluated, indicating stronger semantic grounding. Given SSD300's relatively shallow, single-shot architecture, this suggests that simpler detectors with dense anchor coverage may encourage more localized, object-centered attention patterns. Meanwhile, FCOS and SSDLite320 exhibit the highest

semantic spuriosity. These more complex, anchor-free detectors may suffer from less precise spatial modeling or over-reliance on contextual features, leading to increased activation in background or co-occurring regions and increased semantic spuriosity. Further, LayerCAM and EigenGradCAM seem best suited for evaluating spuriosity in object detectors as they yield the most consistent and varied scores.

4.3 Vision-Language Models (VLMs)

We evaluate CLIP-based VLMs on the COCO-Stuff 10K [4, 22] validation dataset using semantic class labels of each image as text prompts and compute *SSS* for each prediction.

Table 3: Per-image *SSS* results for VLMs on COCO-Stuff 10K [4, 22]

Model	GradCAM ↓	Grad CAM++ ↓	Score CAM ↓	Layer CAM ↓	Eigen CAM ↓	Eigen GradCAM ↓	Ablation CAM ↓
TinyCLIP-8M [57]	0.4902	0.6191	**0.4104**	0.6118	0.8617	0.7524	**0.3909**
TinyCLIP-40M [57]	0.4815	0.6782	0.5442	0.7472	0.6699	0.6255	0.4714
TinyCLIP-61M [57]	0.4745	0.7197	0.5310	0.7506	**0.6667**	0.6244	0.4593
CLIP-ViT-B [38]	0.4748	0.6165	0.6642	0.7819	0.6704	**0.6054**	0.4546
MetaCLIP-400m [59]	**0.4643**	**0.5116**	0.7218	0.6755	0.6820	0.6196	0.5863
DFN [11]	0.4947	0.5399	0.6721	**0.5823**	0.7509	0.7072	0.4677
CLIP-rsicd-v2 [5]	0.4668	0.6201	0.6316	0.8063	0.6765	0.6073	0.4473
QuiltNet-B-32 [21]	0.5255	0.7239	0.6217	0.8834	0.7489	0.6784	0.5535
PLIP [20]	0.5087	0.6768	0.6221	0.8145	0.6965	0.6492	0.5091
FashionCLIP [5]	0.5046	0.7321	0.5570	0.7748	0.6678	0.6455	0.5146
PubMedCLIP [10]	0.5028	0.6114	0.6089	0.7250	0.7055	0.6579	0.5186

From Table 3, we observe that the VLMs studied generally exhibit higher spuriosity than models in other modalities. This likely stems from their reliance on token-based text encodings, which introduce additional abstraction and increase the risk of ambiguous associations between textual prompts and image regions. Unlike supervised classifiers that focus on discrete object categories, VLMs are trained to align global image features with rich, entangled text representations, often resulting in attention to co-occurring background elements or contextually related but semantically irrelevant regions. This suggests that VLMs benefit from multimodal flexibility but are prone to semantic diffusion, causing spurious focus in their visual grounding. Despite these challenges, TinyCLIP-8M and MetaCLIP consistently achieve lower *SSS*, suggesting that lightweight or carefully regularized variants can better maintain semantic alignment. Their compact architectures and curated training regimes may limit overfitting to spurious background signals and scene-level shortcuts.

By contrast, models such as QuiltNet-B-32 [21], FashionCLIP [5], and PLIP [20] show elevated *SSS* likely due to their specialization in domain-specific tasks (e.g., pathology, fashion). When evaluated on general-purpose datasets like COCO-Stuff, these models rely more heavily on contextual or domain-specific priors, leading to greater spuriosity, thus validating *SSS* as a diagnostic tool. Crucially, this serves as an implicit falsifiability test: had FashionCLIP or PLIP achieved lower *SSS* than CLIP variants pretrained on general-purpose datasets like COCO or ImageNet, this would have cast doubt on the utility of *SSS* as a meaningful indicator of semantic alignment. However, *SSS* reliably captured this disparity, supporting the hypothesis that it is sensitive to semantically misaligned reasoning, providing an empirical sanity check on *SSS*.

Interestingly, some models (e.g. TinyCLIP-8M and DFN) show low spuriosity under GradCAM and AblationCAM but exhibit higher scores under EigenGradCAM and LayerCAM. This variability underscores the sensitivity of *SSS* to the choice of explanation method, and hence highlights the value of SemScore in providing a method to derive suitable configuration settings.

4.4 Attention Mapping Methods

We evaluate Vision Transformers (ViTs) trained on ImageNet1K [42] using the ImageNet1K-S [13] dataset, which includes semantic segmentation annotations. For each model, we compute *SSS* using DAAM [27], GAR [14], and Attention Rollout [1].

Table 4: Per-image *SSS* results for ViT attention maps on ImageNet-S [13]

Model	DAAM ↓	GAR ↓	Attention Rollout ↓
DeiT Tiny [51]	0.5143	**0.2856**	**0.4161**
DeiT Small [51]	0.4990	0.3256	0.6462
DeiT Base [51]	0.4983	0.3996	0.8072
ViT Base [38]	0.5219	0.4583	0.8283
ViT Small [38]	0.5605	0.6254	0.7509
FlexiViT Base [3]	**0.4968**	0.3513	0.6062

As shown in Table 4, smaller DeiT variants like DeiT-Tiny and DeiT-Small achieve lower *SSS* than DeiT-Base, particularly under GAR and Attention Rollout, suggesting that smaller models may be less susceptible to semantic spuriosity and more likely to focus on object-centric regions. For example, DeiT-Tiny achieves the lowest *SSS* with GAR (0.2856), indicating strong semantic alignment. This supports prior observations that smaller models can generalize more cleanly by avoiding overfitting to background or co-occurring noise. Interestingly, FlexiViT-Base achieves the overall lowest *SSS* score under DAAM (0.4968) and performs competitively across other methods. Its flexible token aggregation likely promotes stronger spatial grounding, enabling the model to focus on task-relevant regions more effectively than standard ViTs of similar size. In contrast, ViT-Small and ViT-Base show the highest *SSS* scores across nearly all methods, particularly under GAR (0.6254 and 0.4583) and Attention Rollout (0.7509 and 0.8283). These results suggest greater reliance on semantically irrelevant features, possibly due to training configurations or a lack of inductive bias compared to DeiT and FlexiViT variants. Among attention attribution methods, Attention Rollout shows the greatest spread in *SSS* across models, indicating higher sensitivity to attention variations. This suggests that Attention Rollout may be the most discriminative and informative method for detecting spuriosity in ViTs, making it a strong candidate for use with *SSS* in probing transformer representations. Overall, these results highlight that semantic spuriosity is shaped by model size, architecture, and attention behavior, and that method choice (e.g., Attention Rollout vs. DAAM) significantly impacts interpretability outcomes.

4.5 Qualitative Analysis: Interpreting *SSS* Outcomes

In this section, we present two illustrative cases where SemScore is particularly effective at uncovering model weaknesses. A more extensive set of examples is provided in Appendix A to further demonstrate SemScore's value and practical significance.

4.5.1 High-confidence, correct predictions with high *SSS* reveal reliance on spurious features

(a) Model focuses on vegetation instead of the bunting.

(b) Model highlights background regions rather than the parrot.

(c) Model's attention centers on the janitor, not the broom.

Figure 3: Correct image classification predictions by PVT-v2 [56] on ImageNet-S [13] with high *SSS*. SemScore can identify cases where models make correct predictions by relying on spurious features; these cases are overlooked by conventional performance metrics that do not account for saliency.

The examples in Figure 3 illustrate the scenario of correct predictions made with high confidence scores and high *SSS*. This result suggests that although the model is confident in its prediction, the prediction may have been driven by spurious or semantically irrelevant features rather than true task-relevant cues. By surfacing such hidden reasoning flaws, SemScore reveals insidious model

behavior that might otherwise go unnoticed, enabling targeted analysis and intervention to improve robustness and generalization.

4.5.2 Incorrect, low-confidence predictions with high *SSS* highlight spurious reasoning

(a) Water bottle mispredicted as Megalith

(b) Indian Elephant mispredicted as Tusker

(c) Candle mispredicted as Pedestal

Figure 4: Incorrect image classification predictions by PVT-v2 [56] on ImageNet-S [13] with high *SSS*, indicating reliance on spurious features.

Figure 4 shows examples of incorrect predictions with high *SSS*, indicating reliance on spurious or irrelevant features. This reflects mispredictions due to the model's failure to attend to semantically meaningful cues necessary for correct classification. SemScore enables the discovery of such instances where mispredictions are likely to stem from reliance on semantically irrelevant features, offering insight into the underlying model reasoning flaws to facilitate targeted model improvements.

5 Limitations

While broadly applicable, SemScore entails some practical considerations. First, SemScore exhibits dependence on parameters and attribution methods, resulting in *SSS* variability. As shown in Figure 2, inappropriate methods may yield uninformative explanations, reducing *SSS*'s reliability. Similarly, omitting thresholding can produce noisy or diffuse saliency maps (see Appendix C). While the *SSS* variability caused by inherent differences in saliency behavior across models is not a limitation of SemScore itself, it can nevertheless inhibit SemScore's usability. To mitigate this, SemScore includes tools and templates to help users select appropriate configurations with minimal effort. SemScore's parameter sensitivity and optimal configuration for specific models could be explored in future studies to better understand the theoretical impact and contributions of each component to the overall score.

Additionally, SemScore depends on semantic segmentation masks to define task-relevant regions; this dependence will limit SemScore's usability in contexts without labeled data. This is an unavoidable limitation as SemScore relies on semantic labels to measure spuriosity. The use of pre-trained segmentation models such as SAM 2[39] or F-LMM [58] can alleviate this bottleneck by facilitating the generation of pseudo-ground truth semantic labels. Alternatively, synthetic data can be generated by compositing objects onto diverse backgrounds and producing aligned semantic segmentation masks at scale.

Finally, we acknowledge that SemScore's reliance on saliency maps implies that SemScore is ultimately a proxy measure of the internal reasoning process of a model. Due to the absence of absolute ground truth labels for model reasoning in a causal, human-aligned sense, SemScore can at best be understood as a diagnostic measure for model analysis, rather than a definitive explanation of model reasoning. However, by demonstrating consistent *SSS* degradation in models evaluated out-of-domain (e.g., FashionCLIP on COCO) compared to in-domain models, our falsifiable, task-grounded experiments have shown SemScore's efficacy in providing valid insight into model understanding. We have shown that SemScore remains a purposeful and effective framework that provides researchers with an additional vector in understanding semantic alignment and surface model vulnerabilities across diverse architectures and tasks.

6 Discussion

SemScore provides a scalable, quantitative approach to evaluate semantic alignment in vision models, addressing limitations of qualitative XAI methods. By measuring the extent of model attention overlap with semantically relevant regions, SemScore highlights cases where models rely on spurious features—insights not captured by standard metrics alone. Results indicate that semantic spuriosity is influenced more by training regime and architectural design than model size, clearing the path for further exploration. Models like FlexiViT and ConViT generally show stronger alignment, while hierarchical transformers and some VLMs exhibit high spuriosity, due to attention diffusion or domain-specific biases. Multi-level aggregations (per-prediction, per-image, per-class) further enable fine-grained analysis of failure modes and class-specific biases. These capabilities are crucial for high-stakes applications, where correct predictions made for the wrong reasons can undermine trust. Furthermore, findings of the investigation of the relationship between confidence scores and *SSS* highlight that *SSS* provides complementary information beyond confidence scores and cannot be substituted by prediction confidence alone. While SemScore depends on saliency quality and segmentation labels, its modular design supports adaptable configurations and consistent model comparisons. By quantifying semantic spuriosity, SemScore advances explainability beyond visual inspection to support more reliable, interpretable model development.

6.1 Social Impact and Ethical Considerations

Our work enhances the capabilities of researchers by supporting model evaluation and explainability—promoting greater transparency and more semantically coherent models. As an open-source toolkit, SemScore is widely accessible, enabling broad adoption and extension without significant barriers. While the method can be computationally demanding—an expected tradeoff in deep learning workflows—it introduces minimal ethical or social risks. Rather, SemScore serves as a transparent, objective tool for evaluating model behavior, contributing positively to responsible AI development.

6.2 Future Work

Future work could enhance SemScore's breadth and usability. Support for emerging saliency techniques like Attention Guided CAM [26] and perturbation-based methods like LIME [41] and Shapley values [33, 7, 55] may enrich interpretability by offering complementary views of model behavior. Further, SemScore may be extended to feature-level annotations to enable the evaluation of non-spatial spuriosity beyond object boundaries by comparing non-spatial attributions to annotated feature maps.

Beyond vision, SemScore could be adapted to language tasks by analyzing textual attention over labeled text, enabling the assessment of semantic relevance in NLP models. In AI safety and robustness, *SSS* could help detect vulnerabilities by evaluating reasoning under adversarial or distributional shifts. Finally, SemScore outputs could also inform training by serving as auxiliary supervision (e.g., regularizing models toward low-spuriosity regions). Integration into popular explainability libraries like Quantus [17] could further streamline adoption and standardize evaluation across the XAI community.

Acknowledgments and Disclosure of Funding

This research was supported by the Sensemaking and Surveillance Centre of Expertise (S&S CoE) at the Home Team Science and Technology Agency (HTX), Singapore, under the guidance of Koa Ming Di, Keng-Teck Ma, Wang Yadong, Chua Teck Wee, and Christopher Sia. Additional assistance was provided by Cheng Jia Xian from S&S CoE, HTX.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers, 2020. URL `https://arxiv.org/abs/2005.00928`.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps, 2020. URL `https://arxiv.org/abs/1810.03292`.

[3] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes, 2023. URL https://arxiv.org/abs/2212.08013.

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context, 2018. URL https://arxiv.org/abs/1612.03716.

[5] Patrick John Chia, Giuseppe Attanasio, Federico Bianchi, Silvia Terragni, Ana Rita Magalhães, Diogo Goncalves, Ciro Greco, and Jacopo Tagliabue. Contrastive language and vision learning of general fashion concepts. *Scientific Reports*, 12(1):18958, Nov 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-23052-9. URL https://doi.org/10.1038/s41598-022-23052-9.

[6] Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, and Tarek R. Besold. A historical perspective of explainable artificial intelligence. *WIREs Data Mining and Knowledge Discovery*, 11(1):e1391, 2021. doi: https://doi.org/10.1002/widm.1391. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1391.

[7] Ian Covert, Chanwoo Kim, and Su-In Lee. Learning to estimate shapley values with vision transformers, 2023. URL https://arxiv.org/abs/2206.05282.

[8] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021.

[9] Saurabh Desai and Harish G. Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 972–980, 2020. doi: 10.1109/WACV45572.2020.9093360.

[10] Sedigheh Eslami, Christoph Meinel, and Gerard De Melo. Pubmedclip: How much does clip benefit visual question answering in the medical domain? In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1151–1163, 2023.

[11] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023.

[12] Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. A typology for exploring the mitigation of shortcut behavior, 2024. URL https://arxiv.org/abs/2203.03668.

[13] Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *TPAMI*, 2022.

[14] Jacob Gildenblat. jacobgil/vit-explain: Explainability for vision transformers. https://github.com/jacobgil/vit-explain, 2021. Accessed: 2025-07-29.

[15] Syed Nouman Hasany, Caroline Petitjean, and Fabrice Mériaudeau. Seg-xres-cam: Explaining spatially local regions in image segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3733–3738, 2023. doi: 10.1109/CVPRW59228.2023.00384.

[16] Anees Ur Rehman Hashmi, Dwarikanath Mahapatra, and Mohammad Yaqub. Envisioning medclip: A deep dive into explainability for medical vision-language models, 2024. URL https://arxiv.org/abs/2403.18996.

[17] Anna Hedström, Leander Weber, Dilyara Bareeva, Daniel Krakowczyk, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M. C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond, 2023. URL https://arxiv.org/abs/2202.06861.

[18] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019. URL https://arxiv.org/abs/1905.02244.

[19] Qi Huang, Emanuele Mezzi, Osman Mutlu, Miltiadis Kofinas, Vidya Prasad, Shadnan Azwad Khan, Elena Ranguelova, and Niki van Stein. *Beyond the Veil of Similarity: Quantifying Semantic Continuity in Explainable AI*, page 308–331. Springer Nature Switzerland, 2024. ISBN 9783031637872. doi: 10.1007/978-3-031-63787-2_16. URL http://dx.doi.org/10.1007/978-3-031-63787-2_16.

[20] Zhi Huang, Federico Bianchi, Mert Yuksekgonul, Thomas J Montine, and James Zou. A visual–language foundation model for pathology image analysis using medical twitter. *Nature Medicine*, pages 1–10, 2023.

[21] Wisdom Oluchi Ikezogwo, Mehmet Saygin Seyfioglu, Fatemeh Ghezloo, Dylan Stefan Chan Geva, Fatwir Sheikh Mohammed, Pavan Kumar Anand, Ranjay Krishna, and Linda Shapiro. Quilt-1m: One million image-text pairs for histopathology, 2023.

[22] Zhenchao Jin. Sssegmenation: An open source supervised semantic segmentation toolbox based on pytorch. *arXiv preprint arXiv:2305.17091*, 2023.

[23] Joon Sik Kim, Gregory Plumb, and Ameet Talwalkar. Sanity simulations for saliency methods, 2022. URL https://arxiv.org/abs/2105.06506.

[24] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1), March 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-08987-4. URL http://dx.doi.org/10.1038/s41467-019-08987-4.

[25] Phuong Quynh Le, Jörg Schlötterer, and Christin Seifert. An xai-based analysis of shortcut learning in neural networks, 2025. URL https://arxiv.org/abs/2504.15664.

[26] Saebom Leem and Hyunseok Seo. Attention guided cam: Visual explanations of vision transformer guided by self-attention, 2024. URL https://arxiv.org/abs/2402.04563.

[27] Yi Liao, Yongsheng Gao, and Weichuan Zhang. Dynamic accumulated attention map for interpreting evolution of decision-making in vision transformer, 2025. URL https://arxiv.org/abs/2503.14640.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. URL https://arxiv.org/abs/1708.02002.

[30] Zachary C. Lipton. The mythos of model interpretability, 2017. URL https://arxiv.org/abs/1606.03490.

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, page 21–37. Springer International Publishing, 2016. ISBN 9783319464480. doi: 10.1007/978-3-319-46448-0_2. URL http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021. URL https://arxiv.org/abs/2103.14030.

[33] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874.

[34] Mazda Moayeri, Wenxiao Wang, Sahil Singla, and Soheil Feizi. Spuriosity rankings: Sorting data to measure and mitigate biases, 2023. URL https://arxiv.org/abs/2212.02648.

[35] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 International Joint Conference on Neural Networks (IJCNN)*, page 1–7. IEEE, July 2020. doi: 10.1109/ijcnn48605.2020.9206626. URL http://dx.doi.org/10.1109/IJCNN48605.2020.9206626.

[36] Fuseini Mumuni and Alhassan Mumuni. Explainable artificial intelligence (xai): from inherent explainability to large language models, 2025. URL https://arxiv.org/abs/2501.09967.

[37] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I. Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps, 2021. URL https://arxiv.org/abs/2006.03204.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[39] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL https://arxiv.org/abs/1506.01497.

[41] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. URL https://arxiv.org/abs/1602.04938.

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[43] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao Li, and Christoph Feichtenhofer. Hiera: A hierarchical vision transformer without the bells-and-whistles. *ICML*, 2023.

[44] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://dx.doi.org/10.1007/s11263-019-01228-7.

[45] Pratinav Seth and Vinay Kumar Sankarapu. Bridging the gap in xai-why reliable metrics matter for explainability and compliance, 2025. URL https://arxiv.org/abs/2502.04695.

[46] David Steinmann, Felix Divo, Maurice Kraus, Antonia Wüst, Lukas Struppek, Felix Friedrich, and Kristian Kersting. Navigating shortcuts, spurious correlations, and confounders: From origins via detection to mitigation, 2024. URL https://arxiv.org/abs/2412.05152.

[47] Ashley Suh, Harry Li, Caitlin Kenney, Kenneth Alperin, and Steven R. Gomez. More questions than answers? lessons from integrating explainable ai into a cyber-ai tool, 2024. URL https://arxiv.org/abs/2408.04746.

[48] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017. URL https://arxiv.org/abs/1703.01365.

[49] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection, 2019. URL https://arxiv.org/abs/1904.01355.

[50] Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. Sanity checks for saliency metrics, 2019. URL https://arxiv.org/abs/1912.01451.

[51] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021.

[52] Van Binh Truong, Truong Thanh Hung Nguyen, Vo Thanh Khang Nguyen, Quoc Khanh Nguyen, and Quoc Hung Cao. Towards better explanations for object detection, 2023. URL https://arxiv.org/abs/2306.02744.

[53] Kira Vinogradova, Alexandr Dibrov, and Gene Myers. Towards interpretable semantic segmentation via gradient-weighted class activation mapping (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(10):13943–13944, April 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i10.7244. URL http://dx.doi.org/10.1609/aaai.v34i10.7244.

[54] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks, 2020. URL https://arxiv.org/abs/1910.01279.

[55] Shaobo Wang, Hongxuan Tang, Mingyang Wang, Hongrui Zhang, Xuyang Liu, Weiya Li, Xuming Hu, and Linfeng Zhang. Gnothi seauton: Empowering faithful self-interpretability in black-box transformers, 2025. URL https://arxiv.org/abs/2410.21815.

[56] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022.

[57] Kan Wu, Houwen Peng, Zhenghong Zhou, Bin Xiao, Mengchen Liu, Lu Yuan, Hong Xuan, Michael Valenzuela, Xi (Stephen) Chen, Xinggang Wang, Hongyang Chao, and Han Hu. Tinyclip: Clip distillation via affinity mimicking and weight inheritance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21970–21980, October 2023.

[58] Size Wu, Sheng Jin, Wenwei Zhang, Lumin Xu, Wentao Liu, Wei Li, and Chen Change Loy. F-lmm: Grounding frozen large multimodal models. *arXiv preprint arXiv:2406.05821*, 2024.

[59] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data, 2023.

[60] Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features?, 2021. URL `https://arxiv.org/abs/2104.14403`.

A Additional examples of SemScore's utility

A.1 Case 1: *SSS* provides informative scores even when saliency maps are difficult for humans to interpret

Figure 5: Sample image classification predictions made by PVT-v2 [56] on ImageNet-S [13] where the saliency maps are patchy and ambiguous. Visually, it is unclear if the model exhibits spuriosity for these predictions, however, the low *SSS* enables a decisive assessment. SemScore provides an unambiguous quantification of spuriosity, removing the need for subjective qualitative assessments.

In scenarios where the saliency map used for visual explanation is visually ambiguous (e.g. when highlighted regions are patchy and diffuse or extend beyond subject-relevant areas) like the samples in Figure 5, human evaluators may struggle to objectively assess the relevance or spuriosity of the model-identified features. Factors like cognitive bias, fatigue, or inattention can further impede reliable assessment. In such situations, SemScore's *SSS* provides an objective and quantitative means of evaluating spuriosity, thereby reducing ambiguity and supporting objective model reasoning interpretation.

A.2 Case 2: High-confidence, correct predictions with low *SSS* suggest strong semantic alignment

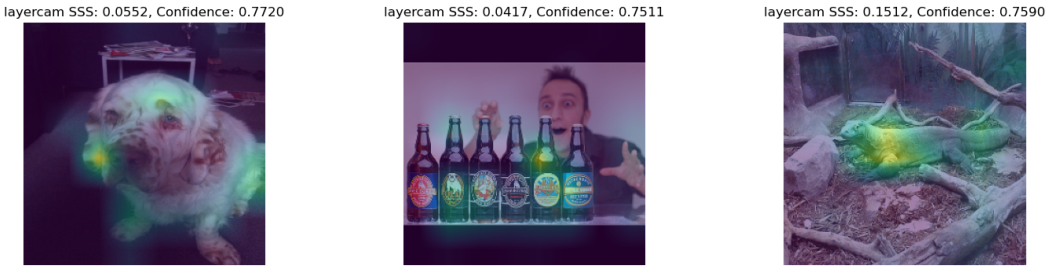

Figure 6: Correct image classification predictions, low *SSS* from PVT-v2 [56] on ImageNet-S [13], reflecting strong semantic alignment.

Examples provided in Figure 6 depict instances where a low *SSS* accompanies a correct and high confidence prediction, indicating strong interpretability as the model's reasoning is well-aligned with semantically relevant features.

A.3 Case 3: Incorrect, low-confidence prediction, with low *SSS* reveal semantically valid but challenging cases.

Figure 7: Sample image classification predictions made by PVT-v2 [56] on ImageNet-S [13] where the predicted class is wrong and the prediction has a low *SSS* indicating low spuriosity. SemScore enables the discovery of such instances where, even though the model appears to correctly identify semantically relevant features, the model nonetheless makes an erroneous prediction.

As illustrated in Figure 7, instances where an incorrect prediction is made with low confidence but also low *SSS* suggest that the model relied on semantically relevant features despite misclassifying the input. This may occur in fine-grained classification tasks where classes are difficult to distinguish due to shared attributes that make them highly similar. For example, while tusks are a valid cue for identifying elephants, a model may still face challenges differentiating between Indian and African elephants, as both elephant classes share this feature and exhibit high visual similarity. This specific case on elephants is explored in Appendix A.4.

A.4 Case 4: Class-based analysis of similar classes

Table 5: Class-based analysis of *SSS* for DeiT Small [51] Image Classification on elephants from the ImageNet-S dataset [13] with EigenCAM.

Class name	***SSS*↓**
Indian elephant	0.5219
African elephant	0.3367

In Table 5, we present a class-level *SSS* analysis of DeiT Small's [51] performance on African and Indian elephant classifications with EigenCAM. The results indicate that DeiT Small demonstrates substantially higher semantic alignment when predicting African elephants. This disparity may reflect an imbalance in training data, such as fewer Indian elephant examples, or a limitation in the model's

ability to differentiate between overlapping features shared by both classes. This example illustrates how SemScore might be used to evaluate class-specific performance or deficiencies in the training regime.

A.5 Case 5: Class-based analysis of top and bottom classes

Table 6: Top-5 classes in terms of *SSS* for DeiT Small [51] image classifier with EigenCAM.

Class name	***SSS*↓**
Waffle iron	0.999
Pinwheel	0.9971
Maze	0.9813
Swing	0.978
Paddlewheel	0.9734

Table 7: Bottom-5 classes in terms of *SSS* for DeiT Small [51] image classifier with EigenCAM.

Class name	***SSS*↓**
Pool table	0.0051
Projectile	0.0026
Warplane	0.002
Dishwasher	0.0019
Plate	0.0005

In Tables 6 and 7, we highlight the ImageNet-S classes [13] for which the DeiT Small [51] classifier exhibits the highest and lowest spuriosity, as measured by *SSS* with EigenCAM. These results demonstrate how *SSS* can be used to uncover training deficiencies and potential model biases at a class-specific level.

A.6 Failure case 1: Limitations of *SSS* under suboptimal saliency methods

Figure 8: Image classification examples illustrating poor SemScore outcomes due to the use of inappropriate saliency map methods for the given task and dataset. Saliency is dispersed across and/or concentrated at semantically irrelevant regions of the image.

When a suboptimal saliency method (e.g., CAM) is applied to a task it is ill-suited for, the resulting explanation may fail to highlight semantically meaningful features. Figure 8 illustrates several such instances; in such cases, SemScore cannot yield a reliable or informative *SSS*, as the input explanation lacks the interpretability necessary for meaningful evaluation.

A.7 Failure Case 2: Limitations of *SSS* with inappropriate saliency map thresholding

As shown in Figure 9, the saliency map correctly highlights the target object but also includes a significant number of pixels in the target object's immediate surroundings. Although the core region of the saliency map corresponds well to the relevant object, the additional surrounding activations introduce noise that undermines the interpretability of the explanation. Without thresholding (i.e., zero threshold), all pixels are treated as equally salient, resulting in an overly diffuse explanation.

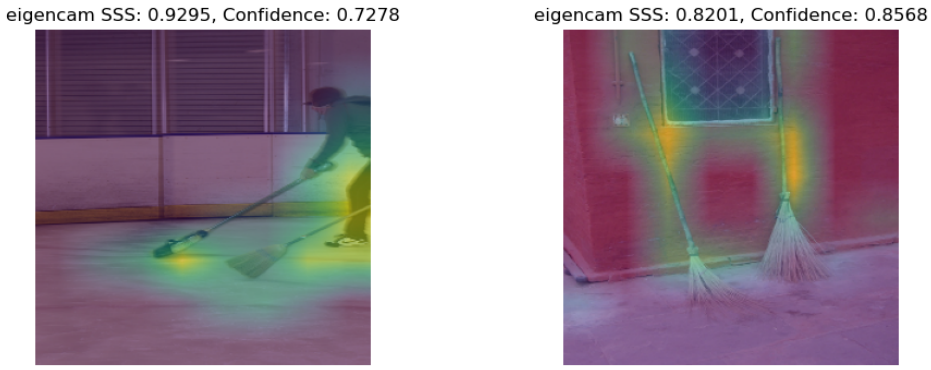

Figure 9: Image classification examples illustrating poor SemScore outcomes due to inappropriate thresholding. Even though the model correctly identified relevant features, the lack of thresholding caused the noisy saliency weights to skew *SSS* upwards, indicating higher spuriosity.

This can hinder the evaluation process by making it harder to distinguish between truly informative features and irrelevant neighboring pixels.

B Code snippets for SemScore's usage in Python

Snippet 1: DataLoader Setup

Code snippet of the Data Loader classes that we provide and can be extended to align datasets to SemScore.

```
from data_loaders import (
    ImageNetSDataLoader,
    COCODataLoader,
    PascalVOCDataLoader,
    ADE20KDataLoader
)

# Users can select existing dataloaders
# Or extend the base DataLoader class for their specific datasets
# Dataloaders are currently implemented for: COCO10k, PascalVOC, ADE20K, ImageNet-S
data_dir = '../data/sssegmentation'
data_loader = ImageNetSDataLoader(data_dir)
```

Snippet 2: Using SemScore

Code snippet depicting sample usage of SemScore to generate prediction-level, image-level, and class-based scores:

```
# Get particular sample
n = 42
image, masks, class_ids, class_labels = data_loader.get_sample(n)

# Instantiate model and target layers
model = timm.create_model('vit_base_patch16_224', pretrained=True)
model.to(torch.device('cuda')).eval()
target_layers = [model.blocks[-1].norm1]

# Alternatively, the SemScore repo provides dozens of model config files
# Within this, models and target layers and prespecified for immediate use
model_config_dir = '../config/vit_config.yaml'
model, target_layers = load_model_config(model_config_dir, model_id='vit_base_patch16
```

```
# Instantiate SemScore with configuration - specifying saliency map method, etc.
# Advanced configuration can be specified or managed using a SemScore configuration
ss = SemScore(
    method="eigencam",
    input_type="class_id",
    model=model,
    target_layers=target_layers
)

# Compute per-prediction scores from SemScore for a given image
scores = ss.compute_single_per_prediction_sss(
    image, masks, class_ids, class_labels, threshold=0.5,
)

# SemScore also computes per-image and per-class scores
per_image_score = ss.compute_per_image_sss(scores)
per_class_score = ss.compute_per_class_sss(scores)

# SemScore can also work with batches of data
# Users can specify sets of indices for the dataloader to return to SemScore
# Currently, batched prediction defaults to all samples in a dataset
indices = [: 24]
data_loader.set_indices(indices)

scores = ss.compute_batch_per_prediction_sss(data_loader, threshold=0.5)
per_image_score = ss.compute_per_image_sss(scores)
per_class_score = ss.compute_per_class_sss(scores)
```

C Pixel importance thresholding

Saliency maps are often noisy, making raw pixel importances unreliable for assessing model spuriosity. As shown in Figure 10, we found that applying a threshold improves *SSS* performance by filtering out low-importance noise.

Soft thresholding involves applying a threshold to the saliency map pixel scores, retaining the original saliency value at pixel (x, y) if it is above the threshold τ, and setting it to 0 otherwise. Thereafter, we compute *SSS* on the remaining pixels. Formally, we define a unified thresholding function $M_\tau^{\text{mode}}(x, y)$, where `mode` $\in$ {`soft`, `hard`}, as:

$$M_\tau^{\texttt{mode}}(x, y) = \begin{cases} M(x, y) \text{ or } 1, & if\ M(x, y) \geq \tau \\ 0, & otherwise \end{cases} \tag{3}$$

This function retains original saliency values in `soft` mode ($M_\tau^{\text{mode}}(x, y)$ = M(x, y)) and binarizes them in `hard` mode ($M_\tau^{\text{mode}}(x, y)$ = 1). In both cases, pixels below the threshold τ are set to zero. The resulting map M_τ^{mode} is then used to compute the *SSS*.

While thresholding improves semantic alignment by suppressing irrelevant activations, it introduces information loss and requires manual tuning based on dataset complexity and saliency map behavior.

Figure 10: Effect of soft thresholding on saliency map sparsity and *SSS*. Increasing the threshold removes low-importance activations, resulting in sparser but more focused maps, highlighting the trade-off between noise reduction and signal preservation.

D Additional results from experiments

Table 8: SemScore Per-prediction *SSS* for ViT Image Classifier Saliency Maps on ImageNet-S [13].

Model	Grad CAM ↓	Grad CAM ++ ↓	Score CAM ↓	Layer CAM ↓	Eigen CAM ↓	Eigen Grad CAM ↓	Ablation CAM ↓	XGrad CAM ↓
DeiT Tiny [51]	0.2048	0.4567	0.3511	0.2639	0.6346	0.6295	0.2592	0.6739
DeiT Small [51]	0.1942	0.4221	0.4550	0.2190	0.4380	0.4691	0.2022	0.6688
DeiT Base [51]	0.3718	**0.4106**	0.5307	0.1970	0.5932	0.6518	0.3279	0.6974
ViT Base [38]	0.2882	0.5653	0.4272	0.2843	0.4058	0.8632	0.3979	0.7455
FlexiViT Base [3]	**0.1590**	0.5845	**0.3215**	0.2358	0.3718	0.2902	0.2450	0.6352
ConViT Small [8]	0.1832	0.4948	0.5064	**0.1627**	0.4992	0.3146	**0.1805**	0.6607
Hiera-Base Plus [43]	0.8055	0.8336	0.5422	0.5412	0.4866	0.3299	0.3028	0.7319
PVT v2 [56]	0.6304	0.6938	0.4499	0.2909	**0.3093**	**0.2243**	0.8270	**0.5472**
Swin Base [32]	0.4074	0.7515	0.5330	0.8010	0.3548	0.2533	0.8923	0.6147

Table 9: SemScore Per-prediction *SSS* for VLM Saliency Maps on COCO-Stuff 10K [4, 22].

Model	Grad CAM ↓	Grad CAM ++ ↓	Score CAM ↓	Layer CAM ↓	Eigen CAM ↓	Eigen Grad CAM ↓	Ablation CAM ↓	XGrad CAM ↓	Full Grad ↓
TinyCLIP 8M [57]	0.5159	0.6386	**0.4259**	0.6593	0.8606	0.7641	**0.4114**	0.8066	**0.7727**
TinyCLIP 40M [57]	0.5119	0.6840	0.5985	0.7981	0.7251	0.6659	0.5022	0.7910	0.8055
TinyCLIP 61M [57]	0.5046	0.7186	0.5883	0.7959	**0.7230**	0.6637	0.4899	0.7906	0.8056
CLIP ViT-B [38]	0.5065	0.6477	0.6769	0.8104	0.7196	**0.6552**	0.4780	0.7990	0.8155
MetaCLIP 400m [59]	**0.4891**	**0.5355**	0.7257	0.7114	0.7295	0.6617	0.5733	0.8083	0.8098
DFN [11]	0.4984	0.5445	0.6805	**0.5865**	0.7606	0.7104	0.4715	0.8080	0.7767
CLIP rsicd-v2 [6]	0.4975	0.6459	0.6499	0.8281	0.7275	0.6543	0.4711	0.7947	0.8090
QuiltNet B-32 [21]	0.5387	0.7280	0.6436	0.8922	0.7708	0.6932	0.5624	0.7821	0.8026
PLIP [20]	0.5435	0.6956	0.6480	0.8342	0.7421	0.6922	0.5367	0.7970	0.7984
Fashion CLIP [5]	0.5245	0.7265	0.5997	0.8119	0.7150	0.6771	0.5332	0.8109	0.8162
PubMed CLIP [10]	0.5331	0.6399	0.6345	0.7585	0.7470	0.6991	0.5448	**0.7854**	0.7723