ACCURACY, ARCHITECTURE, OR METHOD: WHAT DETERMINES XAI QUALITY IN VISION MODELS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Explainable Artificial Intelligence (XAI) is an emerging topic which aims at making opaque AI models' decisions transparent and understandable. Various methods have been proposed to explain deep learning models across different domains. What remains an ongoing debate is the question of how the explanation quality of XAI methods should be evaluated. Numerous quantitative metrics have been suggested in recent literature which aim to foster comparability along certain explanation properties such as *localization*, *faithfulness*, *complexity* and *randomization*. Previous studies, mostly using off-the-shelf pretrained backbones, have indicated that explanations might be influenced by three factors: the *model architecture*, the *model performance*, and the *XAI method*. To what extent, however, still remains unclear. In addition, the interplay between these three factors has also not been investigated yet. Our study contributes to this research gap by determining how much these three factors actually impact the nominal explanation quality of plain image classifiers. We therefore train 26 different Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) from scratch, spanning model sizes from 3.4M to 210M parameters. In our study we use the same training parameters on a 50-class ImageNet subset with pixel-level object masks, and score five XAI methods with twelve evaluation metrics at eight checkpoints per model, covering accuracies from 16% to 91%. From our image classification scenario three results emerge: 1. Accuracy has the largest impact on *localization*. On the other hand, *complexity*, *faithfulness* and *randomization* metrics are mainly influenced by the XAI method and do not significantly improve as the model learns. 2. With ongoing training, perturbation-based XAI methods tend towards outperforming gradient-based methods on transformers regarding most metrics we measure. The opposite behavior can be observed for CNNs. 3. Within a model architecture, it is the normalization layers that have the highest impact on *localization* metrics. Simply increasing the number of model parameters, however, does not yield measurable improvements regarding any XAI metric.

1 INTRODUCTION

Modern vision models for image classification rest on two architectural baselines that continue to coexist. Traditional Convolutional Neural Networks (CNNs) such as VGG (Simonyan & Zisserman,

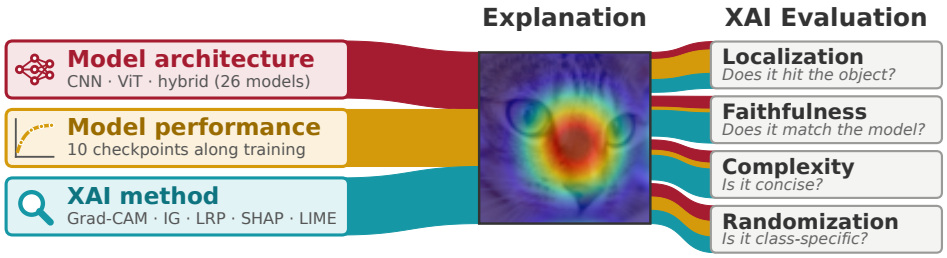

Figure 1: We analyze the impact and interplay of model architecture, performance, and XAI method on explanations, measured by localization, faithfulness, complexity, and randomization metrics.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2015), ResNet (He et al., 2016) and EfficientNetV2 (Tan & Le, 2021) backbones remain in heavy use, specifically when training data is sparse. Modern CNN model families such as ConvNeXt (Liu et al., 2022) already incorporate modern transformer-inspired architecture tweaks. Vision transformers such as Swin (Liu et al., 2021), PVTv2 (Wang et al., 2022), and convolution-attention hybrids such as MaxViT (Tu et al., 2022) have become the modern state-of-the-art models in many Computer Vision (CV) settings, especially when enough training data is available.

What all of these models share is opacity. In high-risk applications, such as medical diagnosis (Agarwal et al., 2023), autonomous driving (Zhao et al., 2025) and industrial applications (Seitz et al., 2024), a prediction that cannot be accounted for is hard to deploy both responsibly and also legally: E.g., the EU AI Act obliges providers of high-risk systems to make them transparent enough for users to interpret and oversee the output (European Union, 2024). XAI aims to answer this demand by extracting explainable evidence for individual decisions (Kalasampath et al., 2025). Several approaches of XAI methods exist. For the task of image classification, *feature attribution* methods remain frequently used, which score each input pixel for its importance to the model's prediction (Wang et al., 2024). These XAI methods are split into *back-propagation* methods and *perturbation* methods. The former propagate gradients or relevance scores backwards through the network and so depend on its internals. Tools such as Integrated Gradients (IG) (Sundararajan et al., 2017), Grad-CAM (Selvaraju et al., 2017), and Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) are part of this category. Perturbation methods treat the model as a black box and fit a local surrogate under occlusion. Therefore, they do not rely on the model's internal layer structure. Popular methods in this category are LIME (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017).

Unlike a model's prediction, explanations usually don't have a "ground-truth" to be checked against, which is why the field has developed various quantitative, functionally-grounded metrics for evaluating explanations given by any XAI method. A standard taxonomy sorts the available metrics into categories (Nauta et al., 2023), four of which are measurable for every method we study: localization, also known as *coherence* (*does the explanation fall on the object depicted?*), faithfulness, also known as *correctness* (*does removing the pixels it calls important change the prediction?*), complexity (*is the explanation concise?*) and randomization (*does it change at all when the explanandum is randomized?*). Literature suggests that the scores such metrics return might be dependent on at least three pillars: the XAI method itself, the architecture it is applied to, and how well that model has been trained (Terzieva et al., 2025; Klein et al., 2024; Knoll et al., 2025). Existing comparisons vary only single factors without combining all three of them at once, leaving a significant research gap in analyzing the actual numerical impact and the bigger picture of the interplay between all three factors on explanation quality (Figure 1). As a consequence, the following research questions emerge:

(RQ1) How and by how much does accuracy impact explanation quality?

(RQ2) How does the choice of XAI method influence the explanation outcome - and how does that choice interplay with the model architecture and performance?

(RQ3) How does the model architecture impact explanations?

Our study design. We aim at answering these research questions by providing the following study design: First, we train from scratch 26 classifier backbones from seven model families. We perform the training with the same setup on the same 50-class ImageNet-S_{50} (Gao et al., 2022) dataset. Then, during training, we save eight checkpoints per model spanning 16% to 91% top-1 accuracy. Because every architecture traverses the same accuracy range under one and the same training regime, the effect of accuracy can be read off for each model, and the three factors can be told apart. On the 208 resulting model states we then obtain explanations from five attribution methods. Finally, we evaluate them with twelve XAI metrics from the four taxonomy categories.

Three findings. Our study reveals three novel key findings connected to our research questions:

1. (RQ1) **Accuracy decides where an explanation points** (Section 4). Whether an explanation is located on the classified object rises with model accuracy for almost all (96%) of the 585 model/method/localization-metric trajectories (median Spearman $\rho = 0.90$). Over the accuracy range we cover it is the largest of the three factors for every localization metric. Outside localization the ordering inverts: The complexity, faithfulness and randomization

of an attribution map barely respond to training at all, and are mainly decided by the XAI method instead.

2. (RQ2) **Perturbation methods overtake generic gradient methods on transformers, and fall behind them on CNNs** (Section 5). When it comes to choosing the best XAI methods for specific model checkpoints, we notice that the perturbation methods LIME and KernelSHAP end up ahead of the gradient methods on most transformers and hybrids and behind them on CNNs at later training stages: at convergence one of them is the best of the available XAI methods on six of nine transformers, but on only one of seventeen CNNs. Both XAI method families start out tied early in training and separate as the models learn. As a result, a certain XAI method which performs best for a converged model is not necessarily the best choice for the same model during early training stages.
3. (RQ3) **Normalization layers matter more than model size** (Section 6). Making a model ten times larger (in terms of parameters) within its family doesn't measurably impact explanation quality. Adding Batch Normalization (BN) to a VGG-based model at identical depth and parameter count improves localization for all five XAI methods, by as much as roughly twenty to forty accuracy points for Grad-CAM and LRP, and fifteen to thirty for the perturbation methods. On the other hand, the LayerNorm-plus-patchify-stem package degrades localization metrics for Grad-CAM and IG, which can be observed in the ConvNeXt and PVTv2 models.

Section 2 places our study in context to prior work. Section 3 describes the study design in greater detail. Sections 4-6 present the three key findings and Section 7 discusses the results and states the limits of the study, before we draw a conclusion in Section 8.

2 Related Work

Transformers and attributions. Applying attribution to Vision Transformers (ViTs) typically requires architecture-specific adaptions. Chefer et al. (2021) present a relevance-propagation rule that traverses the attention operation rather than relying on raw attention weights, and validate it against competing methods on a single converged ViT backbone. Because the backbone is held fixed, their ranking characterizes one model at convergence rather than architectures in general. Achtibat et al. (2024) propose attention-aware LRP rules for LayerNorm, softmax and matrix multiplication, and Jain & Wallace (2019) show that raw attention weights are not faithful explanations by themselves. Both works argue at the level of the attribution method: they establish how an attribution *should be computed* on an attention block, not how the quality of the resulting explanation changes once the backbone or its training state varies. In contrast, we systematically vary both model backbone and training state at a fixed evaluation protocol, scoring five attribution methods on nine transformer and hybrid backbones at eight checkpoints each.

Benchmarking XAI methods. Several recent studies evaluate attribution methods across models and metrics, yet none fully disentangle architecture, training progress and method choice. LATEC (Klein et al., 2024) scores 17 methods against 20 metrics over several datasets and architectures and concludes that no method dominates universally, so the choice must be made per use case. Its metric suite, however, covers only faithfulness, robustness and complexity, while localization is explicitly dismissed as a "niche" criterion because it requires ground-truth masks. In our study, this is the very category in which we measure the strongest effects. Moreover, each of their three image architectures contributes only a single converged model, so the influence of training state is never examined. Knoll et al. (2025) indicate that XAI method quality can depend on the underlying model, yet their scope is restricted to four different CNN models only, precluding general conclusions. Göller et al. (2026) evaluate two XAI methods across ten checkpoints of three CNNs. Their exploratory scope remains narrow and no transformers are included in their study. Hesse et al. (2024) come closest to our setting by treating backbone design as an explicit variable across 23 attribution methods, but they score every method with a single self-proposed metric (the in-domain single-deletion score), which is a deletion-based correctness proxy. Localization, complexity and randomization are never measured, and their accuracy-explainability trade-off is read off converged ImageNet backbones that each contribute one accuracy and one metric score. Their follow-up, Hesse et al. (2026), scale the model selection to 326 off-the-shelf backbones yet entirely omits explainability metrics from their nine model quality dimensions considered. Terzieva et al. (2025) track feature attributions across

training epochs and observe that they are unstable early and stabilize later. However, they study a single architecture family and rely on entropy-style uncertainty rather than explanation quality. Consequently, their finding is about *when* attributions settle, not *how good* they become on *which backbone*.

With the exception of the latter, all of the above works compare converged models with mostly narrow-scoped evaluation scores. Architecture is therefore confounded with the accuracy a given backbone happens to reach, and effects that arise along the training trajectory remain unobserved. Our study closes this research gap by moving every architecture through the same accuracy range under a single training regime. We deliberately choose a multi-category suite of functionally grounded metrics (localization, faithfulness, complexity and randomization) rather than a single score which captures only one narrow aspect of an explanation, providing a far more comprehensive and comparable picture of explanation quality.

3 Study Design

3.1 Data

We use ImageNet-S_{50} (Gao et al., 2022), the 50-class subset of ImageNet (Deng et al., 2009) shipping with human-verified segmentation masks. The presence of these high-quality segmentation masks allows us to seamlessly compute the localization quality of the explanations, thus making this dataset optimal for our experiments. The training set consists of $64\,431$ images. The test split is a fixed, deterministically ordered set of 488 mask-annotated images with a mean mask coverage of 0.39. Grad-CAM, IG and LRP are evaluated on all 488 images, whereas LIME and KernelSHAP are scored on a smaller 200-image subset due to their exhaustive computational resource demands (2048 model forward passes per image required, for each of both methods). We create that subset by taking the first four images of each of the 50 classes, which are also present in the larger 488-image evaluation pool.

3.2 Models and training

Our tested models consist of six VGG– (Simonyan & Zisserman, 2015), four ResNet– (He et al., 2016), three EfficientNetV2– (Tan & Le, 2021) and four ConvNeXt–classifiers (Liu et al., 2022) on the convolutional side, and three Swin– (Liu et al., 2021), three PVTv2– (Wang et al., 2022) and three MaxViT–models (Tu et al., 2022) on the transformer side. Model parameters range from 3.4M to 210M. Within each family, the members differ only in width and depth, which isolates model scale; VGG additionally provides BN variants that differ from their plain counterparts by adding BN to each convolutional layer. Using PyTorch, every model is trained from scratch with the same training regime: AdamW (Loshchilov & Hutter, 2019) (initial learning rate $lr = 3 \times 10^{-4}$, weight decay $w = 1 \times 10^{-4}$) is used as optimizer together with cosine scheduling, a batch size of 256, random-resized-crop to 224×224 pixels and horizontal-flip augmentation, a maximum training duration of 200 epochs and a fixed seed at 42 with deterministic environment settings. We retain eight checkpoints per model spanning the accuracy range as evenly as possible. For each model we cover $\sim$20%-90% top-1 accuracy. Appendix A provides more details for every model individually.

3.3 XAI methods

We evaluate five XAI attribution methods from both backpropagation– and perturbation–based approaches: Grad-CAM (Selvaraju et al., 2017), which weights the final feature maps by their gradients to give a coarse (7×7) and class-discriminative map; IG (Sundararajan et al., 2017), which integrates input gradients along the path to the input pixels; and LRP (Bach et al., 2015) with the ε-plus-flat composite and architecture canonizers (Anders et al., 2026; Kohlbrenner et al., 2020). The perturbation group consists of LIME (Ribeiro et al., 2016) and KernelSHAP (Lundberg & Lee, 2017), both fitted on a 7×7 superpixel grid with 2048 model forward passes per image. An example showing the explanations of all five XAI methods can be seen in Figure 2. We detail our XAI implementations in Appendix B.

Method availability turns out to set minor limitations: LRP has no validated rule set for the specific attention blocks of Swin, PVTv2 and MaxViT as well as for GELU on ConvNeXt yet (Achtibat et al.,

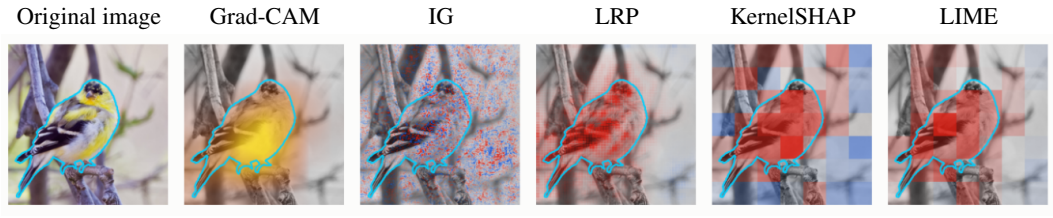

Figure 2: An example image from the Imagenet-S_{50} dataset, explained by five XAI methods on a ResNet50 model. The blue outlines mark the segmentation ground-truth.

2024). We therefore decided to report it for VGG, ResNet and EfficientNetV2 only. Grad-CAM requires a per-architecture reshape transform on the ViTs; there we use the `pytorch-grad-cam` backend (Gildenblat & contributors, 2021) while using Captum's implementation on the other models. All other XAI methods are implemented using Captum (Kokhlikyan et al., 2020).

3.4 Metrics

We use twelve evaluation metrics covering the four taxonomy categories introduced above, based on reference implementations in Quantus (Hedström et al., 2023a). They turn out to be near-independent *across* categories but highly redundant *within*: the five localization metrics rank-correlate at $\rho \geq 0.52$ and the three complexity metrics at $\rho \geq 0.94$ across the 936 (model, checkpoint, XAI method) runs that make up the study (Appendix C). We therefore report one representative per category in the main text.

- **Localization:** *Area Under the Curve (AUC)* treats the attribution as a pixel-wise detector of the ground-truth mask and scores it by Receiver Operating Characteristic (ROC) AUC. It is the probability that a randomly chosen pixel inside the mask receives a higher attribution than a randomly chosen pixel outside it. Therefore its guessing chance level is exactly 0.5 and independent of any image, which is why it is our preferred headline metric. The four other mask-overlap metrics (*Pointing Game*, *Top-K Intersection*, *Relevance Mass*, *Relevance Rank*) have a guessing chance of 0.39, which is the mean mask coverage.
- **Faithfulness:** *Iterative Removal Of Features (IROF)* (Rieger & Hansen, 2020) deletes superpixels in attribution order and reports the area over the resulting class-probability curve.
- **Complexity:** *Sparseness* (Chalasani et al., 2020) is the Gini coefficient of the attribution values. Gini ≈ 0 indicates that every pixel gets roughly equal relevance which results in a diffuse and uninformative heatmap over the whole image. In contrast, Gini $\to 1$ implies that nearly all relevance is sitting on just a handful of pixels, resulting in a concise explanation pointing at very specific areas in the image. *Complexity* and *Effective Complexity* measure the same property as an entropy and a pixel count.
- **Randomization:** *Random Logit* (Sixt et al., 2020) compares the explanation computed for the predicted class against the explanation computed for a randomly drawn other class of the same image. A map that genuinely singles out the class it is supposed to explain should change when that class is replaced, so *low* similarity is good; a score near 1 means the explanation would have looked the same whatever class had been asked about. [1]

4 Finding 1: Accuracy Decides Where an Explanation Points

The first question is which of the three factors (accuracy, architecture, and XAI method) matters most. Figure 3(a) answers it for *localization*: an explanation points at the depicted object roughly in proportion to how well the model classifies it. The trend is positive on 96% of all 585 (model, XAI method, localization-metric) combinations (median Spearman $\rho = 0.90$). Early in training the maps are near guessing chance: averaged over all models, Grad-CAM starts at 0.59 and IG at 0.55 (guessing chance $= 0.50$). Figure 4 illustrates the same progression on individual images.

[1] This is a class-discriminativeness test, not the parameter-randomization sanity check of Adebayo et al. (2018): we compute the latter as eMPRT, but show in Appendix C that on our models it tracks a backbone's normalization rather than its explanation, and therefore base no claim on it.

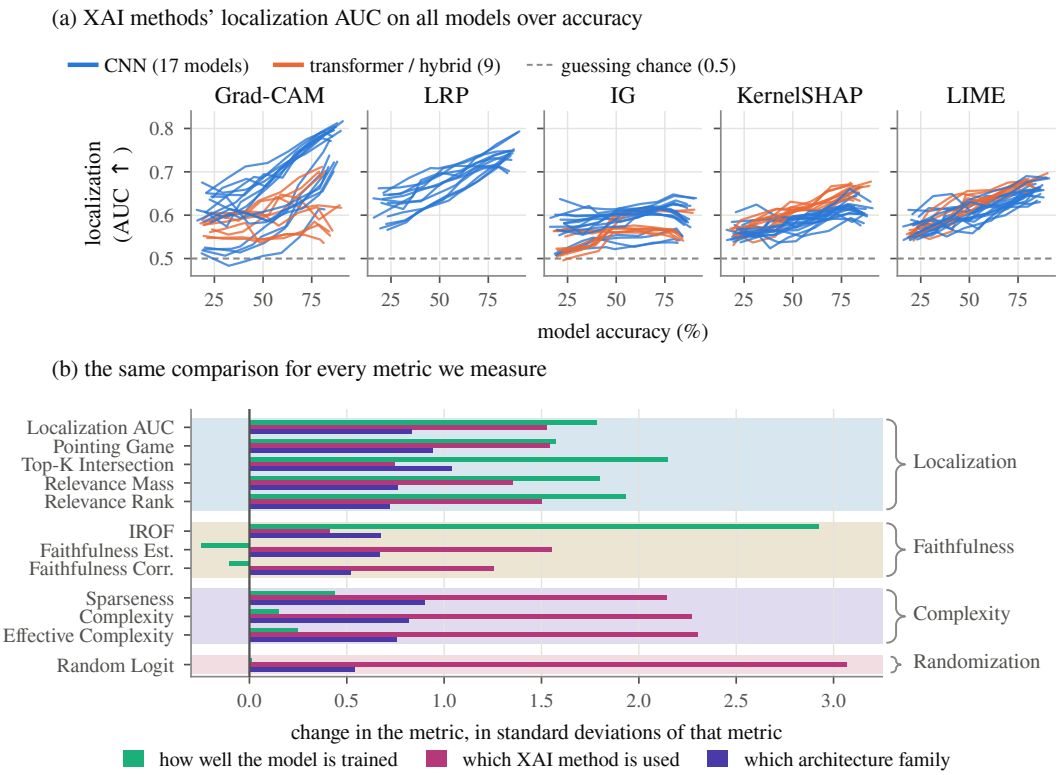

Figure 3: **(a)** One line per model, eight checkpoints each, plotted against the localization metric AUC. CNN models are colored in blue, transformer models in orange. **(b)** For all metrics, we ask how far each factor moves the score on average. In the localization rows at the top, training the model shows the largest impact. The bars are plotted as standard deviations of the metric itself, so they are comparable across metrics. IROF's large accuracy bar is an artefact of its normalization by the unperturbed class probability (footnote 2); the two other faithfulness metrics show even slightly negative correlation with model accuracy.

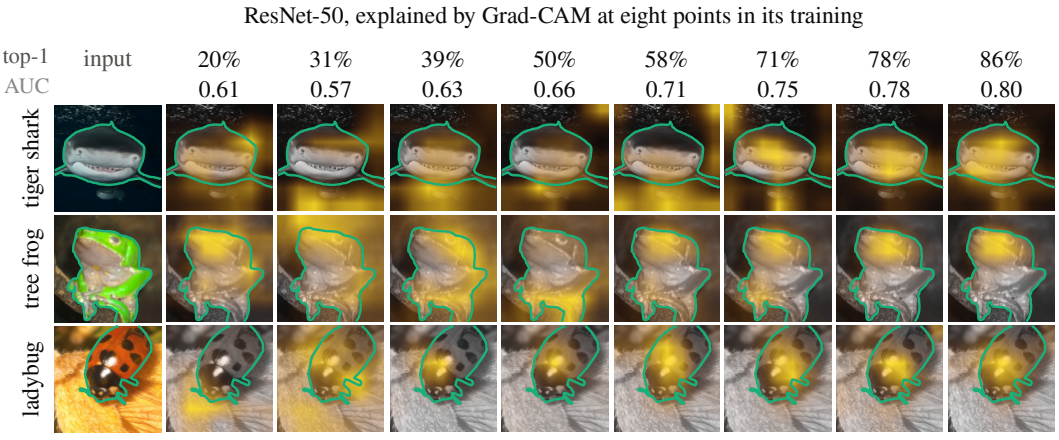

Figure 4: The AUC-accuracy trend on three sample images: ResNet-50 explained by Grad-CAM at each of its eight checkpoints, ordered by accuracy. Yellow colored areas mark positive relevance. The green outline is the ground-truth mask. **top-1** and **AUC** are the model's accuracy and Grad-CAM's localization AUC over the full 488-image evaluation set. Relevance starts spread over the background and concentrates inside the mask as the model learns.

Figure 3(b) extends the comparison to every metric. For localization, accuracy has the largest effect, the XAI method ranks second and the architecture family third. For all other XAI evaluation categories from the taxonomy the ordering inverts: the accuracy bars for Faithfulness Estimate,

Faithfulness Correlation and Random Logit are near zero (or slightly negative), complexity metrics move only a fraction as far, and the choice of XAI method becomes dominant. Conciseness and class-discriminativeness are therefore mainly properties of the attribution algorithm; a better-trained model does not improve them.[2]

Training a model results in *better-localized*, but not overall *better* explanations: accuracy is the most dominant factor that impacts localization, while complexity, faithfulness and class-discriminativeness are mainly influenced by the XAI method used for generating explanations.

5 Finding 2: Perturbation Methods Overtake Generic Gradient Methods on Transformers, and Fall Behind Them on CNNs

Finding 1 treated the three factors of model accuracy, architecture and XAI method separately. However, their interaction is also of interest:

On every checkpoint we subtract the best gradient method (Grad-CAM, IG, LRP) from the best perturbation method (LIME, KernelSHAP). Early in training the two XAI-families are nearly tied; they then start to diverge (Figure 5(a)). The trend is negative on 15 of 17 CNNs and positive on 7 of 9 transformers. At model convergence, when we reach the highest accuracy on the test split, (Figure 5(b)) a perturbation method is best on six of the nine transformers but on only one of the seventeen CNNs. These observations indicate that XAI method performance depends on the interplay between both architecture *and* training stage.

However, the split is not a pure transformer-versus-CNN effect. On all three Swin models Grad-CAM remains narrowly ahead (-0.005 to -0.029 AUC), consistent with Swin's local shifted-window attention preserving spatial structure that Grad-CAM can exploit; PVTv2 and MaxViT mix more globally and favour perturbation methods.

Figure 5(c) shows the complementary view: Grad-CAM improves roughly three times faster on CNNs than on transformers, while LIME and KernelSHAP improve roughly 1.5–$2\times$ faster on transformers. This pattern holds for ten of the twelve metrics.

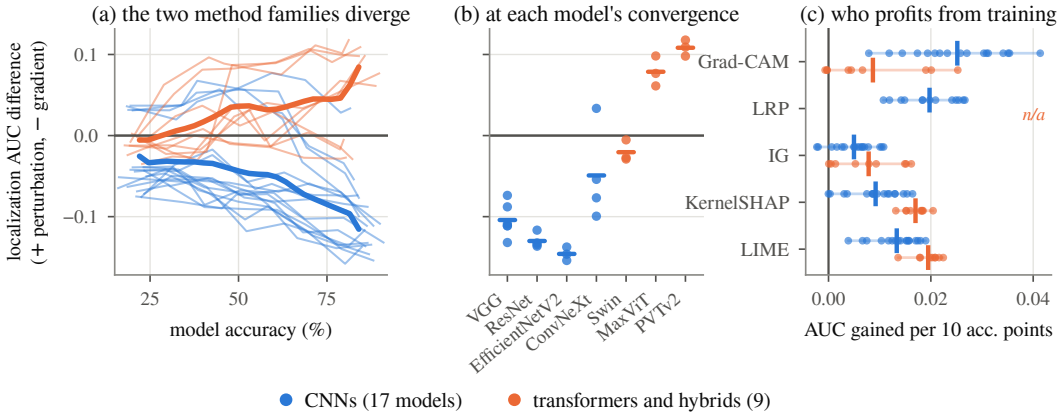

Figure 5: Perturbation vs backpropagation XAI methods - different trends on CNNs and transformers can be seen. **(a)** On each checkpoint, the better score of LIME and KernelSHAP minus the better score of Grad-CAM, IG and LRP; above zero the perturbation methods win. Thin lines are models, thick lines the CNN/transformer model class average. **(b)** The same difference at each model's most accurate checkpoint. **(c)** How steeply each explainer improves along its own model's learning curve.

Which XAI method is the best choice depends on the model's backbone, but it changes while that backbone trains: With ongoing training, perturbation methods win on transformers and hybrids, generic gradient methods on CNNs. An XAI method recommendation for converged models is therefore different from a recommendation for models in early training stages that developers might want to inspect.

[2]IROF is an exception by design: it divides its deletion curve by the unperturbed class probability, which itself grows with accuracy.

6 Finding 3: Normalization Layers Matter More Than Model Size

Two architectural properties are easy to isolate across our model selection: model size (width/depth within a family) and micro-design (layers).

Adding BN at identical depth and parameter count, by contrast, improves localization for all five methods on both VGG depths (Figure 6(a) and Table 4 in Appendix D). The gain is equivalent to 19-37 accuracy points for Grad-CAM and LRP, and 17–31 points for the perturbation methods, and therefore larger than a substantial fraction of the training trajectory itself. This comes along with trade-offs: BN makes Grad-CAM maps more diffuse (lower sparseness) and lowers IROF for four of the five methods at matched accuracy. Additionally, we observe that within each family, a five-fold increase in parameters leaves every XAI metric essentially unchanged (Figure 6(b), circles).

Across our model selection, we observe another micro-design pattern. The ten LayerNorm-plus-patchify backbones (ConvNeXt, Swin, PVTv2) cost Grad-CAM 0.086 and IG 0.055 localization AUC relative to the twelve BN-plus-strided-stem models; KernelSHAP and LIME are unaffected ($+0.020$ and $+0.005$). The gap is already present at a matched 50% top-1 (Figure 7). Only the two XAI methods that propagate through the network are penalized, consistent with the LayerNorm effect reported by Knoll et al. (2025).

A coarse CNN-versus-transformer dichotomy is therefore insufficient: ConvNeXt contains no attention yet inherits the LayerNorm-plus-patchify package and suffers the same XAI degradation for backpropagation methods. The opposite can be oberved for MaxViT, which does contain attention but keeps BN convolutions in early layers.

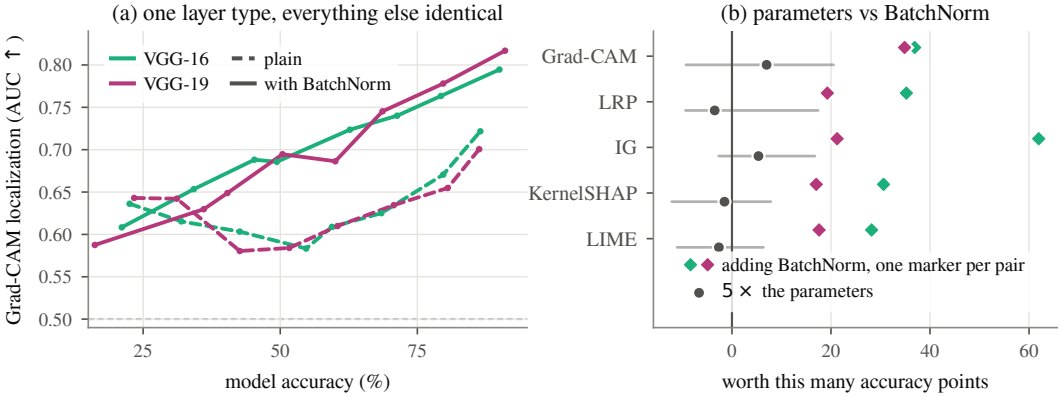

Figure 6: **(a)** Grad-CAM localization AUC on VGG-16 and VGG-19 against their BN twins, which differ from them only by BN layers. Color highlights the depth pair, line style the layer configuration. **(b)** All five XAI methods with both layer configurations in the "currency" of Finding 1: How many accuracy points would be needed to move localization as far. Diamonds are the two VGG pairs, in the colors of (a); circles a five-fold parameter increase within a model family, averaged over all model families. IG's large diamond is a conversion artefact: it barely responds to accuracy, so dividing by its slope inflates the equivalent.

Inside an architecture family, it is the normalization layers, not the model size, that sets the explanation quality: changing one layer type has a remarkably higher impact than an increase of parameters. Additionally, across all models the normalization-plus-stem package acts selectively, penalizing only the XAI methods that propagate through the network.

7 Discussion

Our findings jointly reframe explanation quality. Referring to *RQ1*, we find that accuracy is the dominant driver of localization: maps become object-aligned largely because the model learns where the object is. Complexity, faithfulness and class-specificity, by contrast, are essentially properties of the attribution algorithm and remain largely flat across training. A single scalar "explanation quality" is therefore misleading; localization and the remaining metric families respond to different levers. Architecture and training stage further interact, which answers *RQ2*: Gradient and perturbation

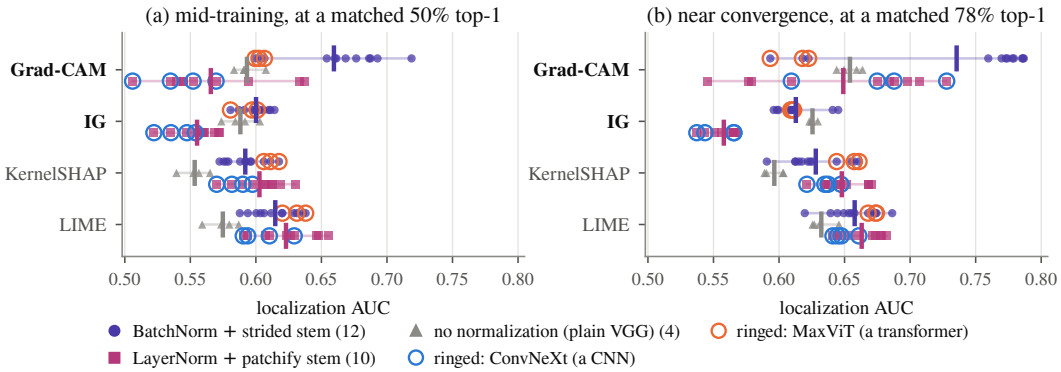

Figure 7: How a backbone's normalization impacts XAI methods. The metric is localization AUC of every backbone at a matched accuracy, one marker per model, **(a)** halfway through training and **(b)** near convergence. Backbones are grouped by what they normalize with: BN with a strided convolutional stem (VGG-BN, ResNet, EfficientNetV2, MaxViT), LayerNorm with a patchify stem (ConvNeXt, Swin, PVTv2), or nothing at all (the four plain VGGs); the vertical tick is the group mean. The two XAI methods that propagate through the network (**bold**) lose on the LayerNorm-plus-patchify backbones at both accuracies; the two that only query the model's outputs never do. LRP has no validated rule set for attention blocks and therefore no LayerNorm entry, so it is left out of the figure. In our model selection, LayerNorm and the patchify stem always occur together, therefore no statement can be made regarding the single use of either of them.

methods start roughly tied and then diverge: the former pull ahead on CNNs, the latter on most transformers and hybrids. An XAI recommendation derived from a converged model doesn't necessarily hold earlier in training. Addressing *RQ3*, we find that parameter count is almost irrelevant within a model family; the decisive factor is the normalization layers. BN improves localization for every method we tested, whereas LayerNorm-plus-patchify selectively degrades methods that propagate through the network. A coarse CNN-versus-transformer dichotomy therefore falls short for analyzing differences between models in XAI quality.

Limitations and Future Work. The study uses a single 50-class ImageNet subset, one training recipe and one seed per architecture. Within-model trends survive seed noise, yet between-architecture comparisons remain single-run estimates. LayerNorm and the patchify stem co-occur throughout the model selection, so only the VGG BN pairs isolate a single micro-architectural factor. Enforcing an identical AdamW schedule without Mixup, CutMix or stochastic depth also places pure transformers at a systematic disadvantage, visible in their lower terminal accuracies.

These limits motivate concrete next steps: multi-seed replications on larger datasets, controlled ablations that orthogonalize normalization from stem design, and extension of the same checkpoint-level protocol to pretrained transformers and to domains beyond natural-image classification.

8 Conclusion

We trained 26 CNNs and vision transformers under a shared regime and scored five XAI methods across their full learning curves. The results show that explanation quality is not a static property of a finished model: localization follows accuracy, while the preferred XAI method depends on architecture, training stage, and micro-architectural choices (especially normalization), which matter more than model size.

These findings have direct implications for both practitioners and XAI developers. Practitioners must select the XAI method carefully. The best choice can change with architecture and with how far the model has been trained. XAI developers, in turn, should evaluate new methods across diverse architectures and multiple accuracy levels rather than on a handful of converged models, as is still common practice.

Ethics Statement

This work involves no human subjects, no personally identifying data and no new data collection. It uses ImageNet-S_{50} (Gao et al., 2022), a public research subset of ImageNet (Deng et al., 2009); the known provenance and representation concerns of ImageNet therefore carry over to our models, and our accuracy figures should not be read as evidence of fitness for any deployed system. The authors declare no competing interests.

Reproducibility Statement

The study is designed to be reproducible. Section 3 details the dataset, the training recipe (optimizer, scheduler, epochs, batch size, augmentation, seed), the checkpoint selection, the five attribution methods with their hyper-parameters, and the twelve metrics. Appendix A lists every model with its parameter count, retained checkpoints and final accuracy. Appendix B details the implementation of the XAI methods. Appendix C reports the redundancy structure of the metric suite, and Appendix D contains the results from our BatchNorm ablation. The training scripts, the evaluation pipeline, and raw metrics evaluated for this paper are released with the supplementary material.

Disclosure of AI Use

In accordance with the ICLR 2027 policy on AI use, we disclose the following. A large language model (Anthropic's Opus 5, used through the Claude Code interface) assisted with parts of this work. It did not originate the research.

What came from the authors. The research questions, the study design, the choice of model families, attribution methods and evaluation metrics, the interpretation of the results and every scientific claim in this paper are the authors' own.

Software. Most code for this study was written with AI assistance and reviewed by the authors: The training scripts for the 26 backbones; the evaluation pipeline that applies the five attribution methods and scores them with Quantus, including the architecture-specific components it required (the Grad-CAM reshape transforms for the transformer families, an LRP canonizer for EfficientNetV2's squeeze-excitation blocks) and several corrections to upstream Quantus behavior that are documented in the released code; and the analysis and plotting code that turns the raw metric logfiles into the figures and tables of this paper.

Analysis of results. The model was used to compute and tabulate the aggregate statistics reported here and to help surface candidate patterns in a large result set. The authors checked the reported quantities against the underlying result files and decided which patterns were substantive enough to report.

Writing. The model produced a first draft of the manuscript outline and of the figure captions from author-supplied results, and was used to tighten formulations. The authors rewrote that draft; the text as submitted is the authors'.

We have reviewed all AI-assisted work and take full responsibility for the content of this paper, including its correctness and its attribution of prior work.

REFERENCES

Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. AttnLRP: Attention-aware layer-wise relevance propagation for transformers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Deevyankar Agarwal, Mario Alberto Berbís, Antonio Luna, Vivian Lipari, Julien Brito Ballester, and Isabel de la Torre-Díez. Automated medical diagnosis of Alzheimer's disease using an EfficientNet convolutional neural network. *Journal of Medical Systems*, 47(1):57, 2023.

Christopher J. Anders, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Software for dataset-wide xai: From local explanations to global insights with zennit, corelay, and virelay. *PLOS ONE*, 21, 01 2026.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10:1–46, 07 2015.

Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1383–1391, 2020.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), 2024. Official Journal of the European Union, L series, 12 July 2024.

Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7457–7476, 2022.

Jacob Gildenblat and contributors. PyTorch library for CAM methods. `https://github.com/jacobgil/pytorch-grad-cam`, 2021.

Kilian Göller, Haadia Amjad, Steffen Seitz, Carsten Knoll, and Ronald Tetzlaff. How a model's architecture and performance influences explainability: A study of Grad-CAM and CRAFT in CNN-based image classification. In *Explainable Artificial Intelligence (xAI), 4th World Conference*. Springer, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Anna Hedström, Leander Weber, Dilyara Krakowczyk, Dina Bareeva, 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. *Journal of Machine Learning Research*, 24(34):1–11, 2023a.

Anna Hedström, Leander Weber, Sebastian Lapuschkin, and Marina Höhne. Sanity checks revisited: An exploration to repair the model parameter randomisation test. In *XAI in Action: Past, Present, and Future Applications (NeurIPS Workshop)*, 2023b.

Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. Benchmarking the attribution quality of vision models. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 37, 2024.

Robin Hesse, Doğukan Bağcı, Bernt Schiele, Simone Schaub-Meyer, and Stefan Roth. Beyond accuracy: What matters in designing well-behaved image classification models? *Transactions on Machine Learning Research (TMLR)*, 2026.

Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 3543–3556, 2019.

K. Kalasampath, K. N. Spoorthi, Sibi Sajeev, Sameer Sankar Kuppa, K. Ajay, and Angulakshmi Maruthamuthu. A literature review on applications of explainable artificial intelligence (XAI). *IEEE Access*, 13:41111–41140, 2025.

Lukas Klein, Carsten Lüth, Udo Schlegel, Till Bungert, Mennatallah El-Assady, and Paul Jäger. Navigating the maze of explainable AI: A systematic approach to evaluating methods and metrics. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 67106–67146, 2024.

Carsten Knoll, Julian Ullrich, Thomas Manjooran, Kilian Göller, Haadia Amjad, Ronald Tetzlaff, and Steffen Seitz. XAIEV – a framework for the evaluation of XAI-algorithms for image classification. In *Explainable Artificial Intelligence (xAI), 3rd World Conference*, pp. 250–263. Springer, 2025.

Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. Towards best practice in explaining neural network decisions with LRP. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2020.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch. *arXiv preprint arXiv:2009.07896*, 2020.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. *ACM Computing Surveys*, 55(13s):1–42, 2023.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Laura Rieger and Lars Kai Hansen. IROF: a low resource evaluation metric for explanation methods. *arXiv preprint arXiv:2003.08747*, 2020.

Steffen Seitz, Thomas Götz, Christopher Lindenberg, Ronald Tetzlaff, and Stephan Schlegel. Towards generalizable classification of partial discharges in gas-insulated hvdc systems using neural networks: Protrusions and particles. *IEEE Transactions on Power Delivery*, 39(3):1491–1499, 2024.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why many modified BP attributions fail. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 9046–9057, 2020.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3319–3328, 2017.

Mingxing Tan and Quoc Le. EfficientNetV2: Smaller models and faster training. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 10096–10106, 2021.

Elena Terzieva, Maximilian Muschalik, Paul Hofman, and Eyke Hüllermeier. Identifying trends in feature attributions during training of neural networks. In *Uncertainty meets Explainability Workshop, ECML-PKDD 2023*, pp. 360–366. Springer, 2025.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. MaxViT: Multi-axis vision transformer. In *European Conference on Computer Vision (ECCV)*, pp. 459–479, 2022.

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):415–424, 2022.

Yongjie Wang, Tong Zhang, Xu Guo, and Zhiqi Shen. Gradient based feature attribution in explainable AI: A technical review. *arXiv preprint arXiv:2403.10415*, 2024.

Jingyuan Zhao, Yuyan Wu, Rui Deng, Susu Xu, Jinpeng Gao, and Andrew Burke. A survey of autonomous driving from a deep learning perspective. *ACM Computing Surveys*, 57(10):1–60, 2025.

A THE MODELS

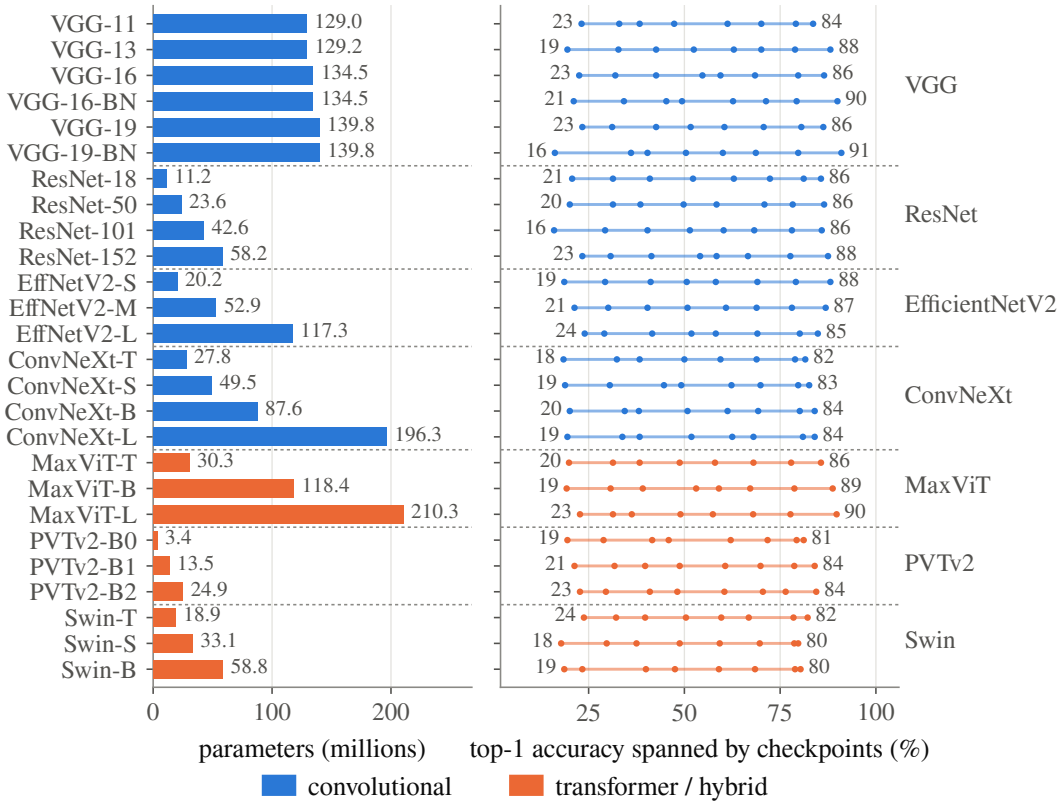

Figure 8: The 26 backbones, grouped by family and ordered by size within each family. Blue marks convolutional models, orange transformer and hybrid models. **Left:** parameter count; the model selection spans a factor of 62, from 3.4M for PVTv2-B0 to 210M for MaxViT-L. Within VGG, the `bn` variants have the same depth and the same parameter count as their plain counterparts, which is what makes them a clean normalization ablation. **Right:** the top-1 accuracy range spanned by each model's retained checkpoints, with one dot per checkpoint; the numbers give the lowest and highest. Every model is sampled from roughly 20% to its own convergence, so every architecture is observed over the same span of accuracies.

Table 1: Every model in the study. "Ckpts" is the number of retained checkpoints and "Top-1" the accuracy of the last one.

Class	Model	Params (M)	GFLOPs	Ckpts	Top-1 (%)	Attribution methods
CNN	VGG-11	129.0	15.2	8	83.6	all five
	VGG-13	129.2	22.6	8	88.1	all five
	VGG-16	134.5	30.9	8	86.5	all five
	VGG-16-BN	134.5	31.0	8	90.0	all five
	VGG-19	139.8	39.3	8	86.3	all five
	VGG-19-BN	139.8	39.4	8	91.0	all five
	ResNet-18	11.2	3.6	8	85.7	all five
	ResNet-50	23.6	8.3	8	86.5	all five
	ResNet-101	42.6	15.7	8	85.9	all five
	ResNet-152	58.2	23.2	8	87.5	all five
	EffNetV2-S	20.2	5.8	8	88.1	all five
	EffNetV2-M	52.9	10.9	8	86.9	all five
	EffNetV2-L	117.3	24.8	8	84.8	all five
	ConvNeXt-T	27.8	8.9	8	81.6	no LRP
	ConvNeXt-S	49.5	17.4	8	82.6	no LRP
	ConvNeXt-B	87.6	30.7	8	84.0	no LRP
	ConvNeXt-L	196.3	68.8	8	84.0	no LRP
Transformer	MaxViT-T	30.3	10.7	8	85.7	no LRP
	MaxViT-B	118.4	46.7	8	89.3	no LRP
	MaxViT-L	210.3	85.5	8	90.2	no LRP
	PVTv2-B0	3.4	1.1	8	81.1	no LRP
	PVTv2-B1	13.5	4.1	8	84.0	no LRP
	PVTv2-B2	24.9	7.8	8	84.4	no LRP
	Swin-T	18.9	6.0	8	82.2	no LRP
	Swin-S	33.1	11.5	8	79.7	no LRP
	Swin-B	58.8	20.4	8	80.3	no LRP

B ATTRIBUTION METHODS: TARGET LAYERS AND HYPER-PARAMETERS

This appendix states exactly where in each of the 26 backbones every Explainable Artificial Intelligence (XAI) method was attached and with which arguments it was called, so that a reader can reconstruct any of the 936 explanation runs from the model name, the checkpoint and the method name alone. Table 2 gives the Grad-CAM target layer per family and Table 3 the arguments of all five methods.

Conventions shared by all five methods. Explanations are computed on the same 224×224 centre crop the models are evaluated on (`Resize(256)` followed by `CenterCrop(224)`). Each input channel is normalized with the ImageNet channel statistics,

$$\boldsymbol{\mu} = (0.485,\, 0.456,\, 0.406)^\top, \quad \boldsymbol{\sigma} = (0.229,\, 0.224,\, 0.225)^\top. \tag{1}$$

The evaluation seed is set to 42, as in training.

Table 2: The Grad-CAM attachment point of every backbone, given as the attribute path on the constructed model; the count in brackets is how many models of that family share it. VGG, ResNet, EfficientNetV2, ConvNeXt and Swin are the `torchvision` implementations, MaxViT and PVTv2 the `timm` ones. "Grid" is the spatial resolution of that layer's output at 224×224 input, verified by a forward hook on each of the 26 models; the channel count varies within a family and is not listed.

Family	Grad-CAM target layer	Module type	Grid
VGG (6)	`features`: last `Conv2d`	`Conv2d`	14×14
ResNet (4)	`layer4[-1]`	residual block	7×7
EfficientNetV2 (3)	`features[-1]`	`Conv2dNormActivation`	7×7
ConvNeXt (4)	`features[-1]`	last stage (`Sequential`)	7×7
Swin (3)	`features[-1][-1]`	`SwinTransformerBlock`	7×7
MaxViT (3)	`stages[-1].blocks[-1].attn_grid`	`PartitionAttentionCl`	7×7
PVTv2 (3)	`stages[-1].norm`	`LayerNorm`	7×7

Grad-CAM. Grad-CAM is the only method whose attachment point is architecture-specific, and Table 2 lists it for every model family.

Integrated Gradients. Integrated Gradients (IG) attributes at the input and needs no layer. We integrate over 32 steps with the trapezoidal rule from the zero baseline, against Quantus' `explain()` wrapper, which hard-codes 10 steps and exposes no batching control. The path integral is evaluated in chunks of 16 steps.

LRP. Layer-wise Relevance Propagation (LRP) is applied through `zennit` (Anders et al., 2026) rather than Captum, whose LRP implementation covers only Conv/Linear/ReLU/Pool/Batch Normalization (BN). The composite is ε-plus-flat (Kohlbrenner et al., 2020) with $\varepsilon = 10^{-6}$: the z^+ rule on the convolutional body, the flat rule on the input layer and the ε rule on the dense layers. Each family contributes its own canonizer. VGG and ResNet use `zennit`'s stock `VGGCanonizer` and `ResNetCanonizer`. EfficientNetV2 additionally requires one we implemented: `SqueezeExcitation.forward` computes $\mathrm{scale}(x) \cdot x$, a product of two *activation* branches for which no relevance rule exists, so relevance is multiplied rather than redistributed and grows by roughly ten orders of magnitude across the $40+$ MBConv blocks. LRP therefore runs on the 13 VGG, ResNet and EfficientNetV2 backbones and not on the 13 ConvNeXt, Swin, MaxViT and PVTv2 ones.

LIME and KernelSHAP. Both are attached to the model as a black box and see no layer at all. Both use the same fixed 7×7 superpixel grid (49 coalition features of 32×32 pixels, identical for every image, model and checkpoint) with the zero baseline as the off-coalition value and 2048 samples per image, drawn 8 coalitions per forward pass. Captum's coalition methods explain one image at a time, so a single explanation costs 2048 forward passes, which is why these two methods are scored on the 200-image stratified subset rather than the full 488 (Section 3). LIME's surrogate is Captum's `SkLearnLasso` with $\alpha = 0.01 \cdot r/5.5$, where r is the model's mean logit range; the rescaling is recomputed at every call rather than once per checkpoint, because the randomization

Table 3: Every non-default argument of the five attribution methods. Settings are identical across all 26 backbones and all eight checkpoints of each; the only per-architecture quantities in the study are the Grad-CAM target layer and reshape transform of Table 2 and the LRP canonizer.

Method	Implementation	Parameters
Grad-CAM	Captum `LayerGradCam` / `pytorch-grad-cam`	target layer per Table 2; ReLU on the weighted activation sum (`relu_attributions=True`); bilinear upsampling to 224×224
IG	Captum `IntegratedGradients`	$n_{\text{steps}} = 32$; `riemann_trapezoid`; zero baseline; `internal_batch_size`$= 16$
LRP	`zennit` `EpsilonPlusFlat`	$\varepsilon = 10^{-6}$; canonizers `VGGCanonizer` (VGG), `ResNetCanonizer` (ResNet), `SequentialMergeBatchNorm` + squeeze-excitation canonizer (EfficientNetV2); per-sample $\max \lvert \cdot \rvert$ rescaling
LIME	Captum `Lime`	7×7 feature mask (49 coalition features of 32×32 px); 2048 samples; zero baseline; 8 perturbations per forward pass; surrogate `SkLearnLasso`, $\alpha = 0.01$ rescaled to the model's logit range
KernelSHAP	Captum `KernelShap`	7×7 feature mask; 2048 samples; zero baseline; 8 perturbations per forward pass; Captum's default SHAP kernel weighting

metrics hand in a perturbed copy of the model whose logit scale differs again. KernelSHAP keeps Captum's default SHAP kernel weighting and needs no such correction, as it solves a weighted least-squares problem with no penalty.

Coverage. Grad-CAM, IG, LIME and KernelSHAP run on all 26 backbones at all eight checkpoints; LRP runs on 13. This gives $4 \times 26 \times 8 + 13 \times 8 = 936$ explanation runs, each scored with twelve metrics.

C Metric Redundancy and the Accuracy Trend

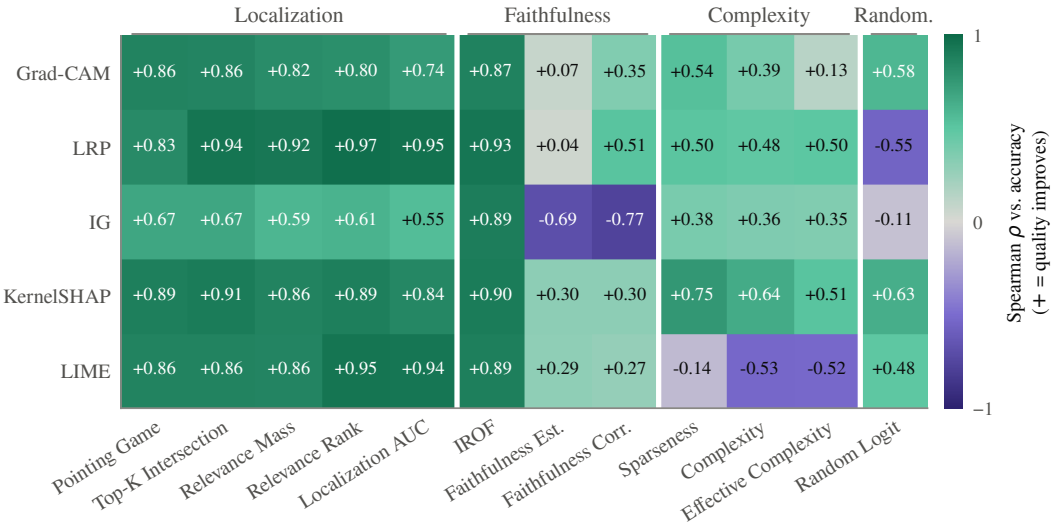

Figure 9: Median Spearman correlation between model accuracy and each metric over the 26 models, signed so that positive always means "explanation quality improves". Localization (and IROF, whose scale grows with model confidence (Section 4)) rises everywhere.

The metrics' behavior is not independent. Figure 10 shows their Spearman rank correlation across the 936 (model, checkpoint, XAI method) runs of the study, with every metric oriented so that larger

means better. Each correlation is therefore taken over 936 paired scores, and the grid covers thirteen computed metrics (the twelve reported throughout plus eMPRT) or 12,160 scores in total, the eight missing entries being LRP's failed eMPRT runs on EfficientNetV2-L. Three observations drive the metric selection of Section 3. First, the five localization metrics form a tight block ($\rho = 0.52$–0.97); Localization Area Under the Curve (AUC) and Relevance Rank Accuracy are near-duplicates ($\rho = 0.97$). Second, the three complexity metrics are effectively one metric ($\rho \geq 0.94$). Third, the four categories are almost orthogonal to each other.

The figure also includes one metric we compute but do not report anywhere else, eMPRT (Hedström et al., 2023b). It *anti*-correlates with the other randomization metric ($\rho = -0.45$), so the two standard ways of asking "does this explanation depend on the model" disagree about which methods pass: Grad-CAM is by far the most class-discriminative explainer in the study (Random Logit 0.20–0.33 at matched accuracy, against 0.72–0.87 for LRP) yet its eMPRT is at or below zero on 22 of the 26 models, while LRP's reaches $+1.40$. Grad-CAM's eMPRT moreover tracks a single architectural property rather than the explanation: it is clearly positive on exactly the four models that contain no normalization layer at all – the plain VGGs, $+0.28$ to $+0.65$ – and at or below $+0.05$ on all 22 that contain BatchNorm or LayerNorm, including VGG-16-BN and VGG-19-BN at -0.30 and -0.31. That is what one would expect if restoring activation scale after randomization blunts a test that measures how much noisier an explanation becomes. We therefore treat eMPRT as uninterpretable on our model selection and base no claim on it; the twelve remaining metrics are the ones reported throughout.

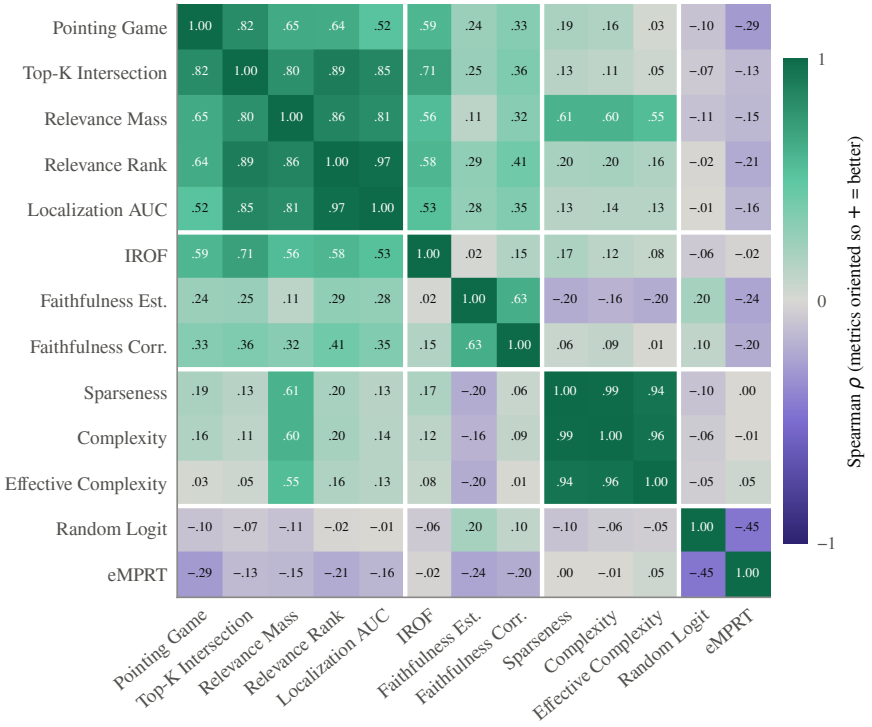

Figure 10: Spearman rank correlation between all thirteen computed metrics across the 936 (model, checkpoint, method) runs, oriented so that + always means better. White lines separate the four categories. eMPRT, rightmost, is shown for completeness and excluded from the paper's results.

D BatchNorm Results

Table 4: The BatchNorm ablation across metric categories. Each entry is the BatchNorm variant minus its plain counterpart with accuracy held fixed by regression over all eight checkpoints of both models, so positive means the BatchNorm model scores higher at equal accuracy. Localization improves for every XAI method and both depths. Grad-CAM is the one method that looses in conciseness: its Sparseness falls and its Effective Complexity rises on both pairs. Directions of improvement: AUC, IROF and Sparseness higher is better; Effective Complexity and Random Logit lower is better.

Pair	Metric	Grad-CAM	LRP	IG	KernelSHAP	LIME
VGG-16 → VGG-16-BN	Localization AUC	$+0.072$	$+0.070$	$+0.038$	$+0.036$	$+0.044$
	IROF	$+1.21$	-2.20	-3.69	-3.34	-2.15
	Sparseness	-0.066	$+0.012$	$+0.038$	$+0.010$	$+0.015$
	Effective Complexity	$+7914$	$+621$	-3789	-213	-1173
	Random Logit	-0.002	$+0.035$	$+0.101$	$+0.001$	$+0.044$
VGG-19 → VGG-19-BN	Localization AUC	$+0.068$	$+0.038$	$+0.013$	$+0.020$	$+0.027$
	IROF	$+2.84$	-0.62	-1.09	-2.10	-1.22
	Sparseness	-0.091	-0.049	$+0.012$	$+0.003$	-0.018
	Effective Complexity	$+8040$	$+7105$	$+863$	-60	$+768$
	Random Logit	-0.008	-0.007	$+0.046$	$+0.003$	$+0.008$