Appendix

Table of Contents

A. Further Experimental Details

A.1. Faithfulness Evaluation Metrics

Modern literature favors evaluations for input attribution methods that are collectively called faithfulness, which intuitively measures how well the attribution scores reflect the true contribution of each input feature to the target output. Although several metrics have been proposed to quantify faithfulness, we adopt the most comprehensive approach, which involves computing the area under the curve (AUC) for the deletion and insertion operations, considering the changes in accuracy and the target probability [13, 26, 46, 49].

The deletion accuracy curve is obtained by progressively removing input features in order of decreasing attribution scores and measuring the model's accuracy at each step. A faithful attribution method should result in a steep drop in performance as the most important features are removed first. The deletion accuracy scores are normalized using the formula $100 - x$, where x is the original score, so that higher scores always indicate better performance.

Similarly, the deletion AOPC curve is generated by gradually removing input features in order of decreasing attribution scores and evaluating the change in the target output probability at each step. A faithful attribution method should lead to a rapid decrease in the target probability as the most important features are removed first.

Using the input images $\mathbf{x}_i$, the perturbed input $\tilde{\mathbf{x}}_i^k$ is created by deleting k% of the most significant patches from $\mathbf{x}_i$. Subsequently, Area Over the Perturbation Curve [18, AOPC] evaluates the mean alteration in the predicted class probability across the entire validation dataset using the following formula:

$$\text{AOPC}(k) = \frac{1}{N} \sum_{i=1}^{N} p\left(\hat{y}|\mathbf{x}_i\right) - p\left(\hat{y}|\tilde{\mathbf{x}}_i^k\right).$$

Here, N represents the total number of instances, $\hat{y}$ stands for the predicted class, and $p(\hat{y}|\cdot)$ denotes the probability of the predicted class.

Conversely, the insertion accuracy curve is generated by gradually adding input features in order of decreasing attribution scores and evaluating the model's performance at each step. A faithful attribution method should lead to a rapid increase in performance as the most important features are added first.

Likewise, the insertion AOPC curve is obtained by progressively adding input features in order of decreasing attribution scores and measuring the change in the target output probability at each step. A faithful attribution method should result in a steep increase in the target probability as the most important features are added first. Similar to deletion accuracy, the insertion AOPC scores are also normalized using $100 - x$, ensuring that higher scores consistently represent better performance.

A.2. True Token Masking

Instead of simply overlaying a color mask, we choose to completely exclude the masked patches from the model's input [15]. At the same time, we preserve accurate positional encodings for the unmasked patches. We term this strategy *True Token Masking*. The conventional method of using the color black (or simply zeroing the tokens in text-based Transformers) for patch masking encounters several issues:

- If a patch is predominantly black, painting it black does not effectively eliminate its informational content. For instance, a black drawing on a white background would remain mostly unchanged.
- Patches might serve computational functions, such as acting as a scratchpad for the model's internal processes. Masking these with black does not prevent the model from using them for such purposes.
- Introducing a black mask can create artifacts in the image, potentially leading to out-of-distribution data, which affects the model's performance.

B. Is SkipPLUS's Choice of Skipping the First Half of the Network Optimal?

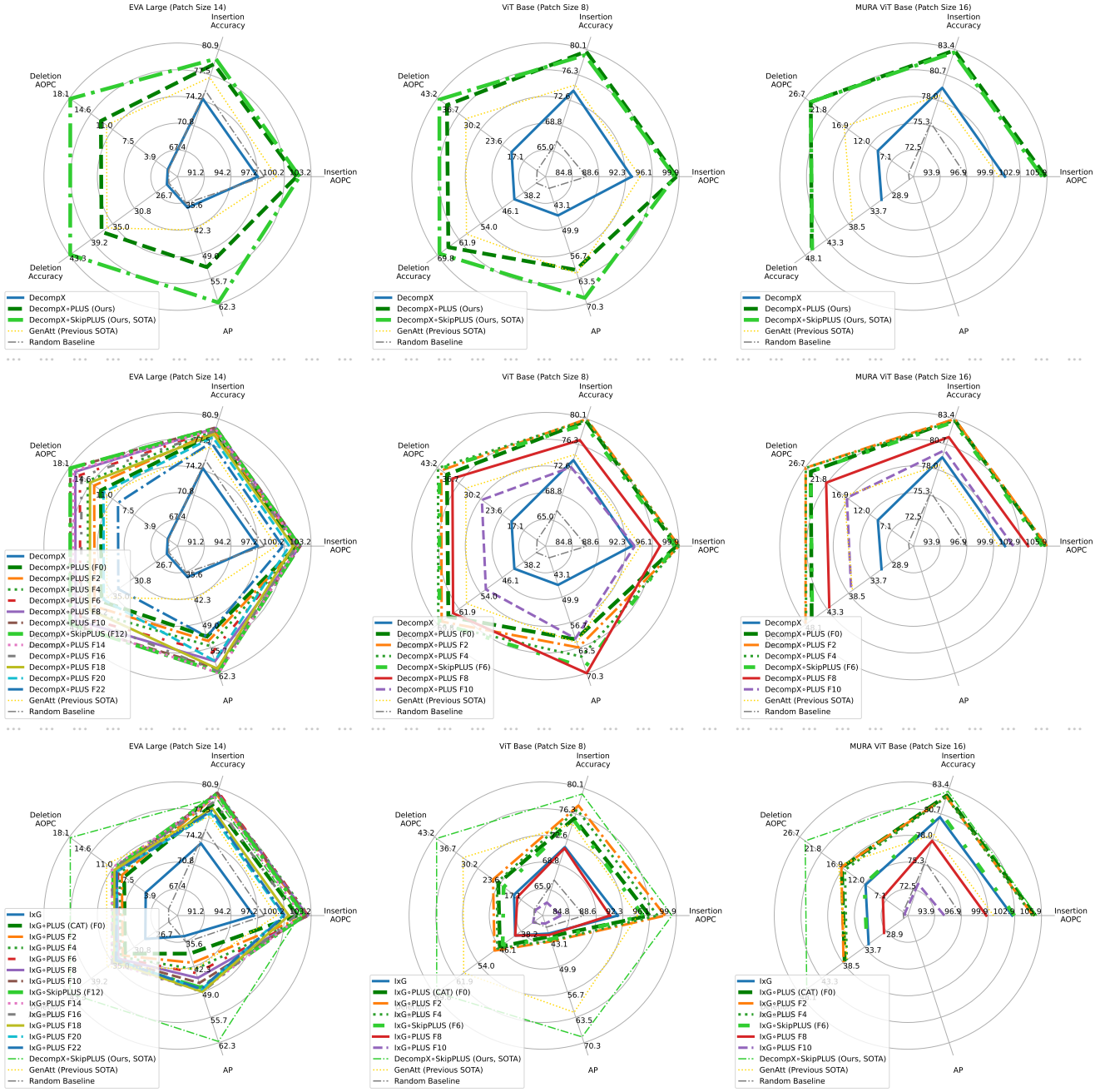

Figure 7. Either PLUS (starting the attribution method from all layers and aggregating information) or SkipPLUS (starting the atribution method from layers in the latter half of the network and aggregating information) are always near the optimal selection of the cutoff layer.

C. Composing PLUS and SkipPLUS With Other Methods

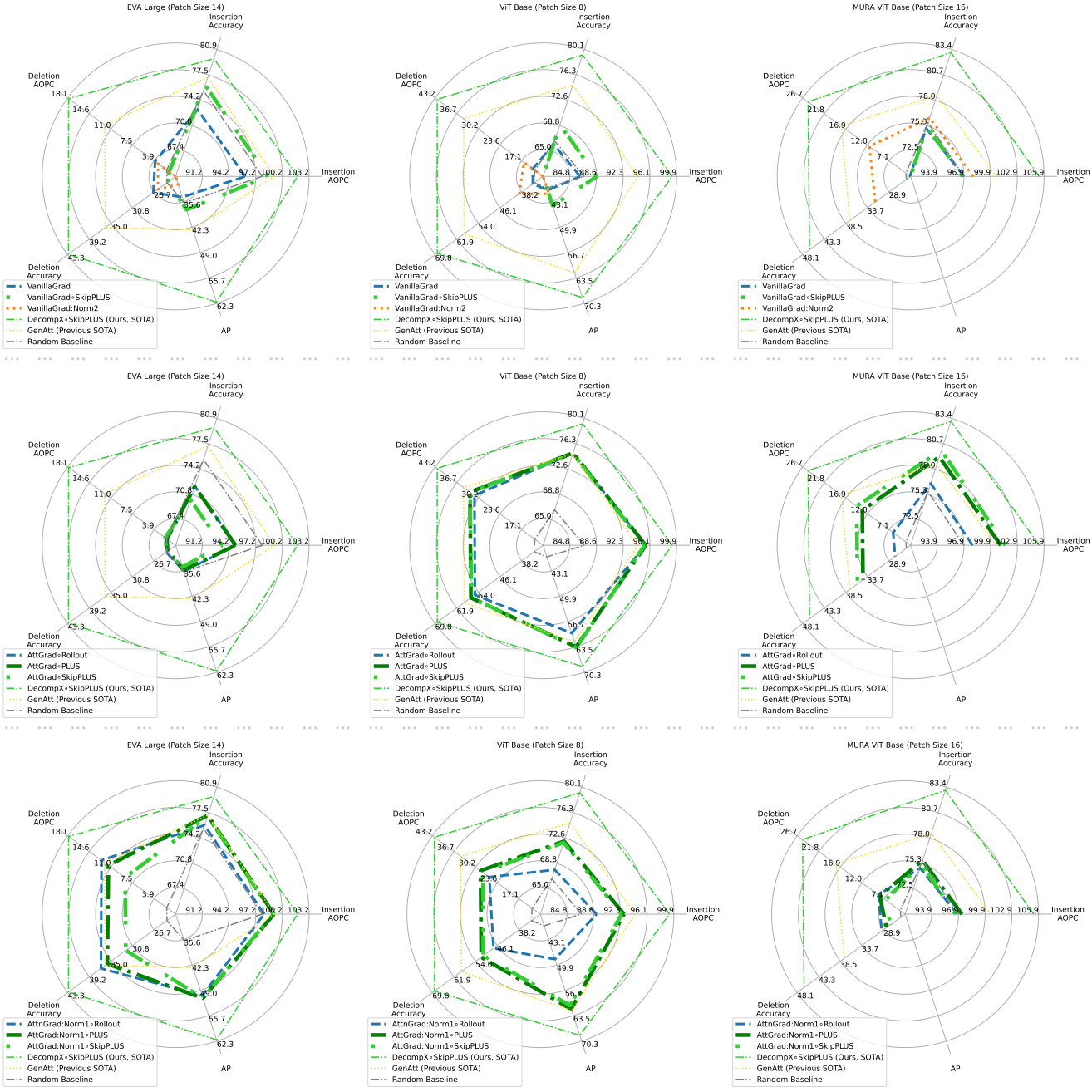

Figure 8. Evaluating the composition of PLUS and SkipPLUS with the Vanilla Gradients of tokens and attention weights.

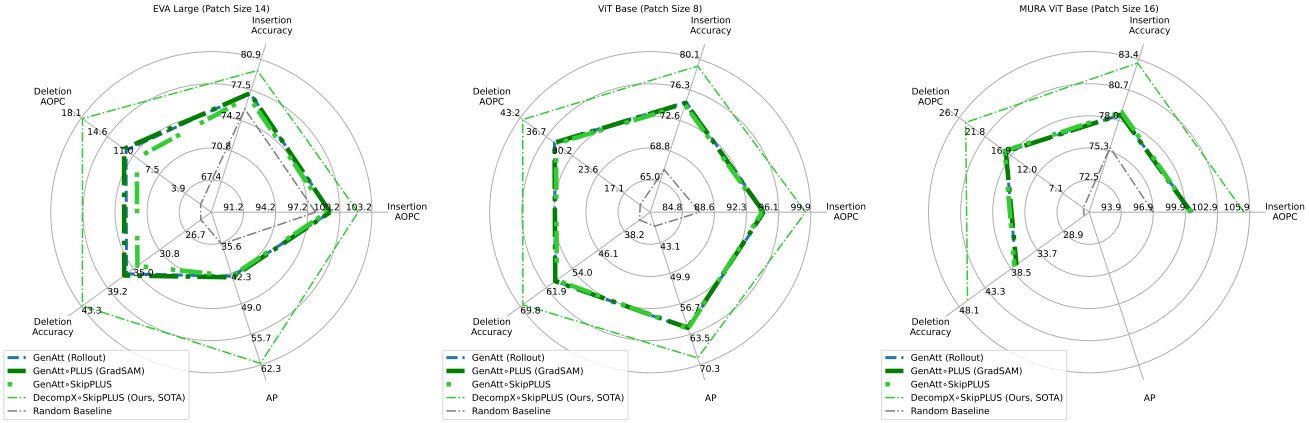

Figure 9. GenAtt, the previous state-of-the-art baseline, is not enhanced by our proposed PLUS or SkipPLUS methods, but it is also not harmed.

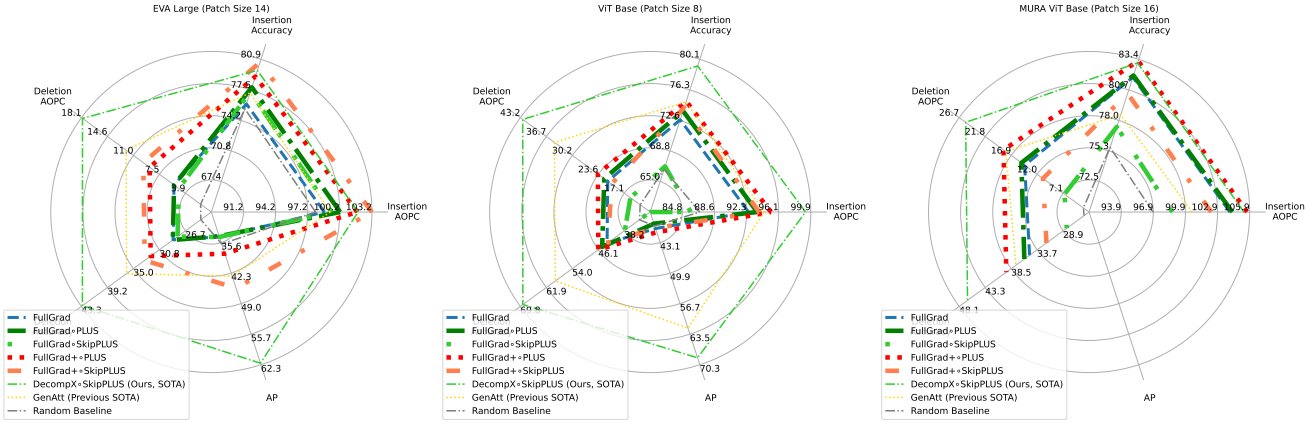

Figure 10. Evaluating the composition of PLUS and SkipPLUS with FullGrad.

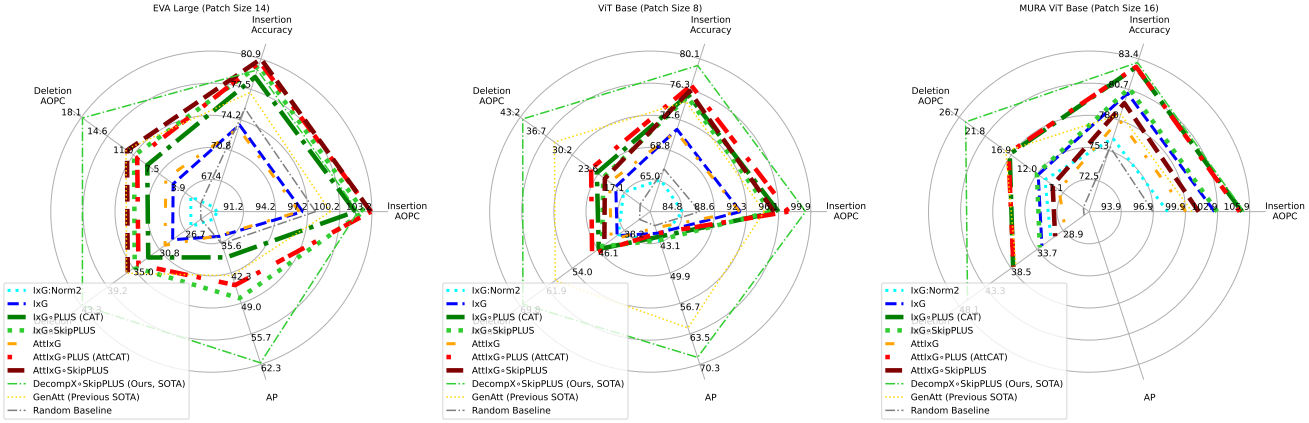

Figure 11. Evaluating the composition of PLUS and SkipPLUS with InputxGradient (IxG) and an attention-enhanced variant, AttIxG. The PLUS compositions of these methods are also known as CAT and AttCAT, respectively.

Figure 12. Evaluating the composition of PLUS and SkipPLUS with CAM methods from CNNs.

D. Qualitative Results

D.1. ViT Base (Patch Size 8)

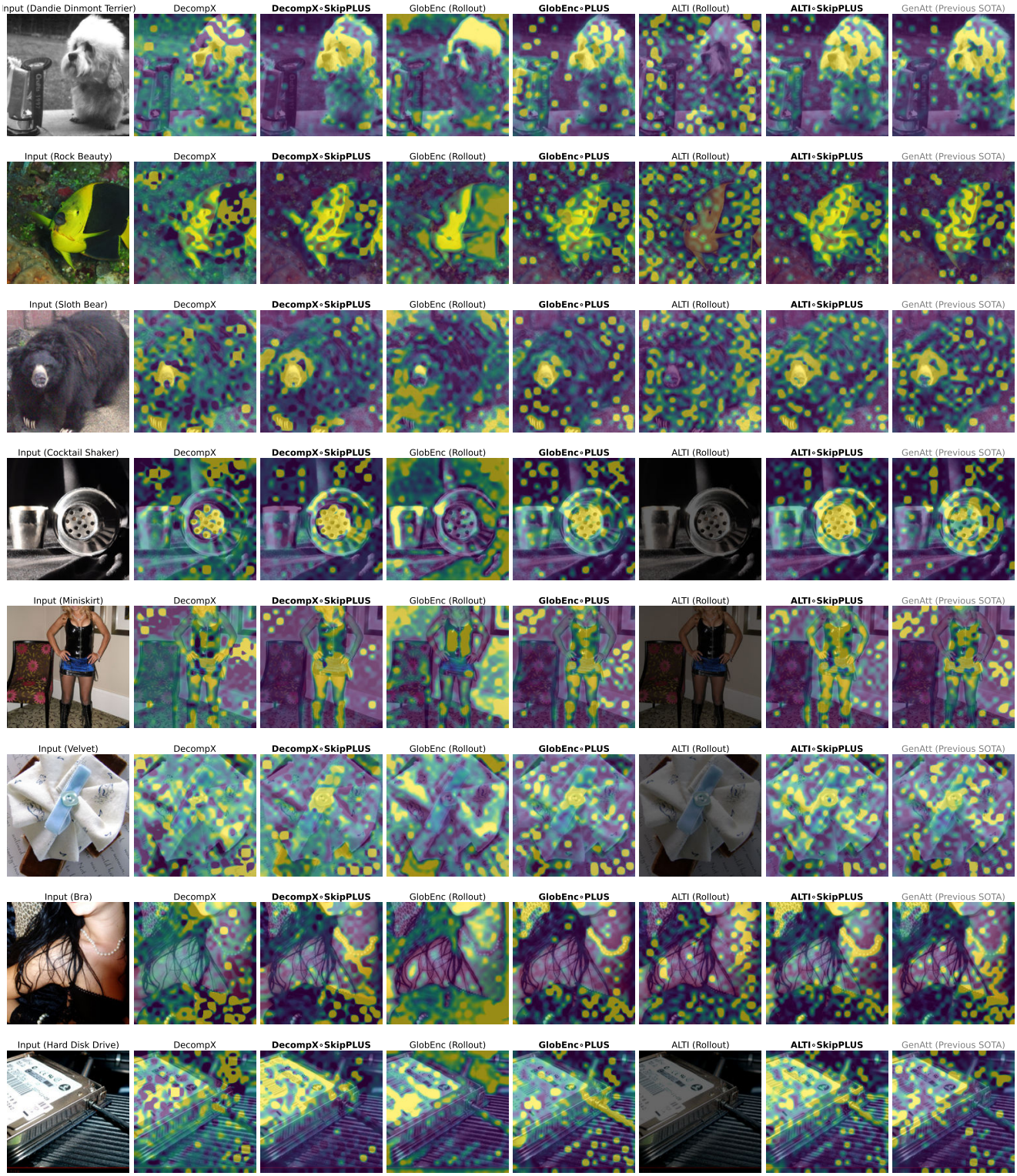

Figure 13. Additional qualitative examples demonstrating the application of SkipPLUS on ViT Base (Patch Size 8). The images presented were selected at random.

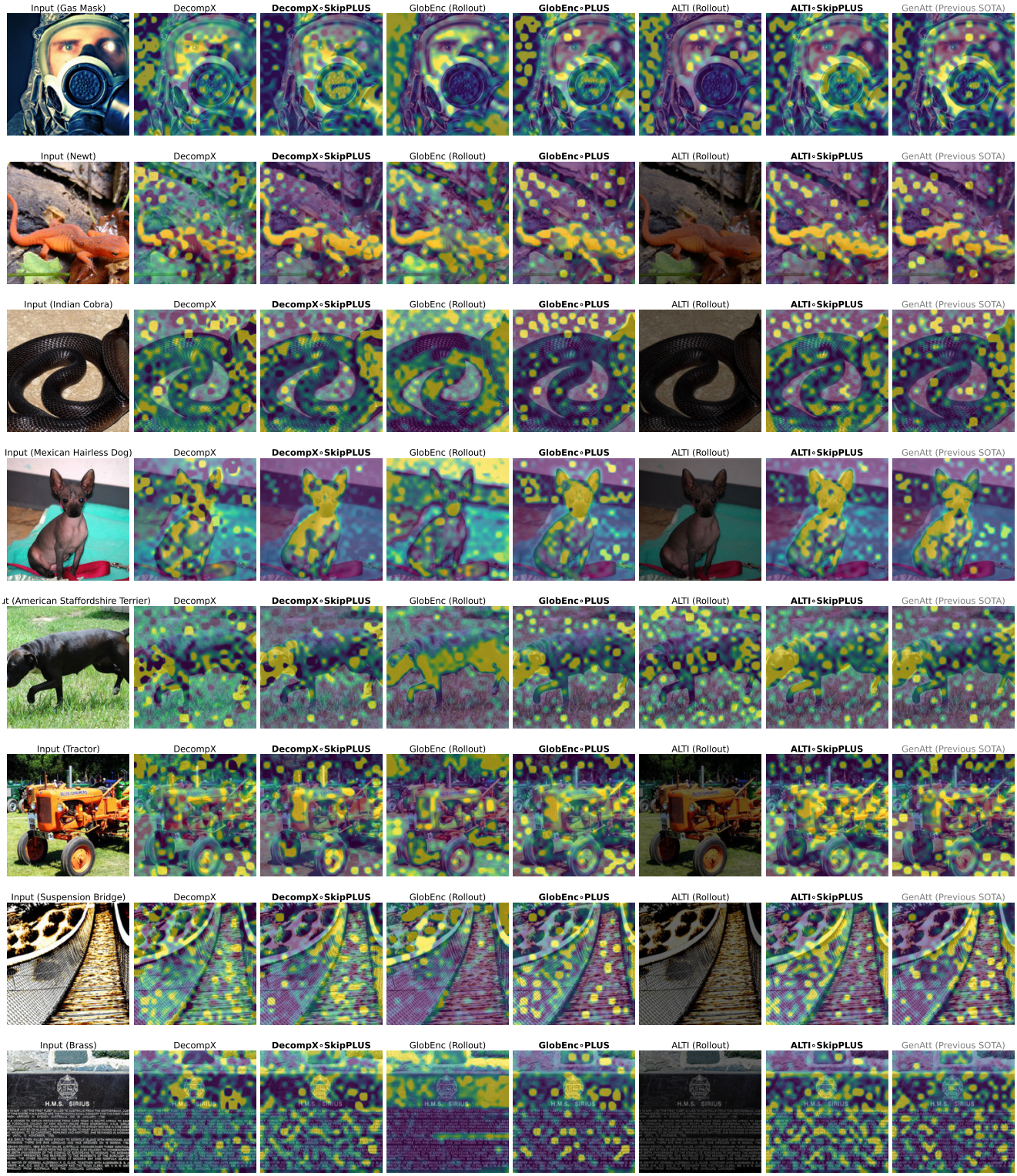

Figure 14. Additional qualitative examples demonstrating the application of SkipPLUS on ViT Base (Patch Size 8). The images presented were selected at random.

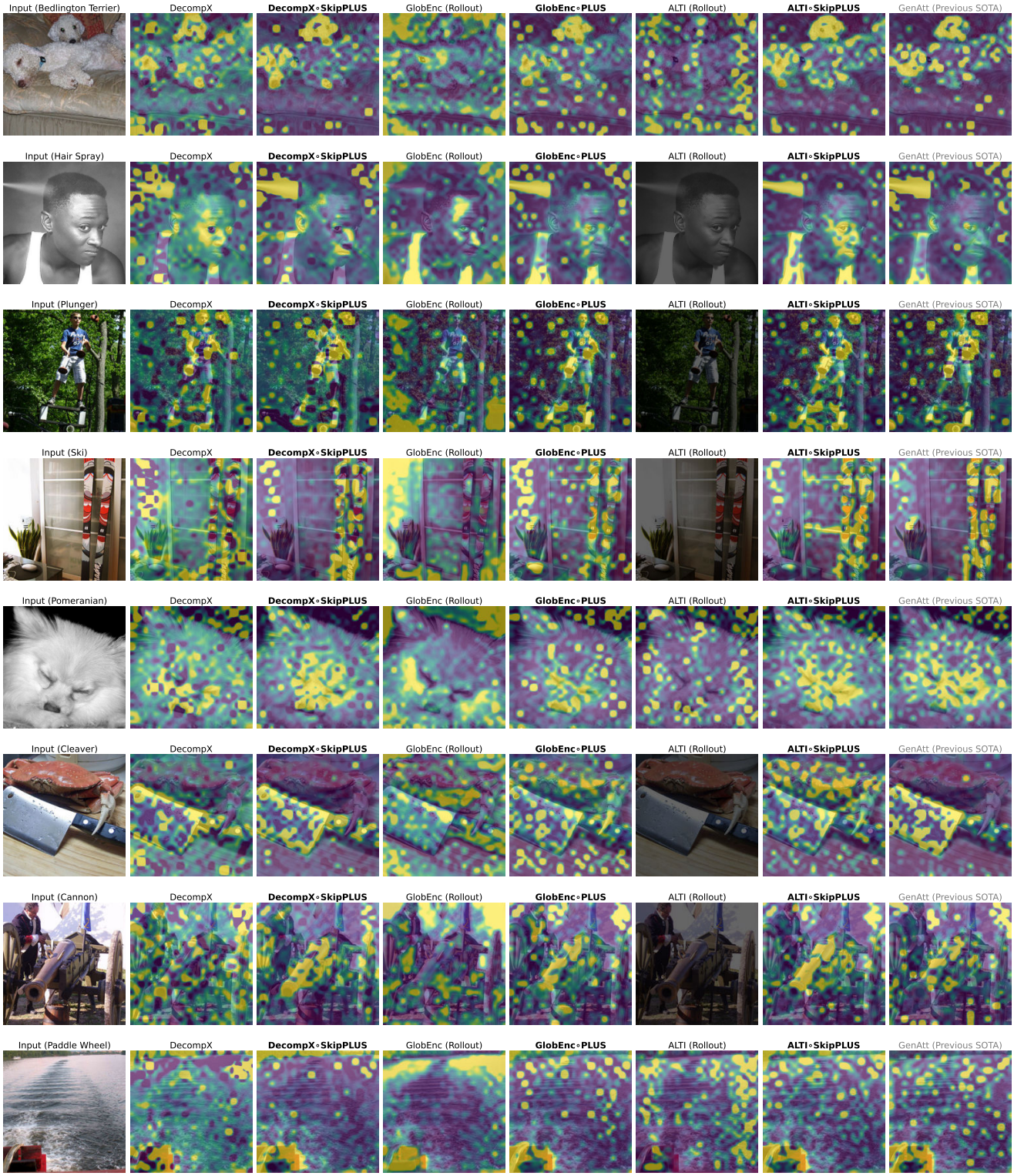

Figure 15. Additional qualitative examples demonstrating the application of SkipPLUS on ViT Base (Patch Size 8). The images presented were selected at random.

Figure 16. Additional qualitative examples demonstrating the application of SkipPLUS on ViT Base (Patch Size 8). The images presented were selected at random.

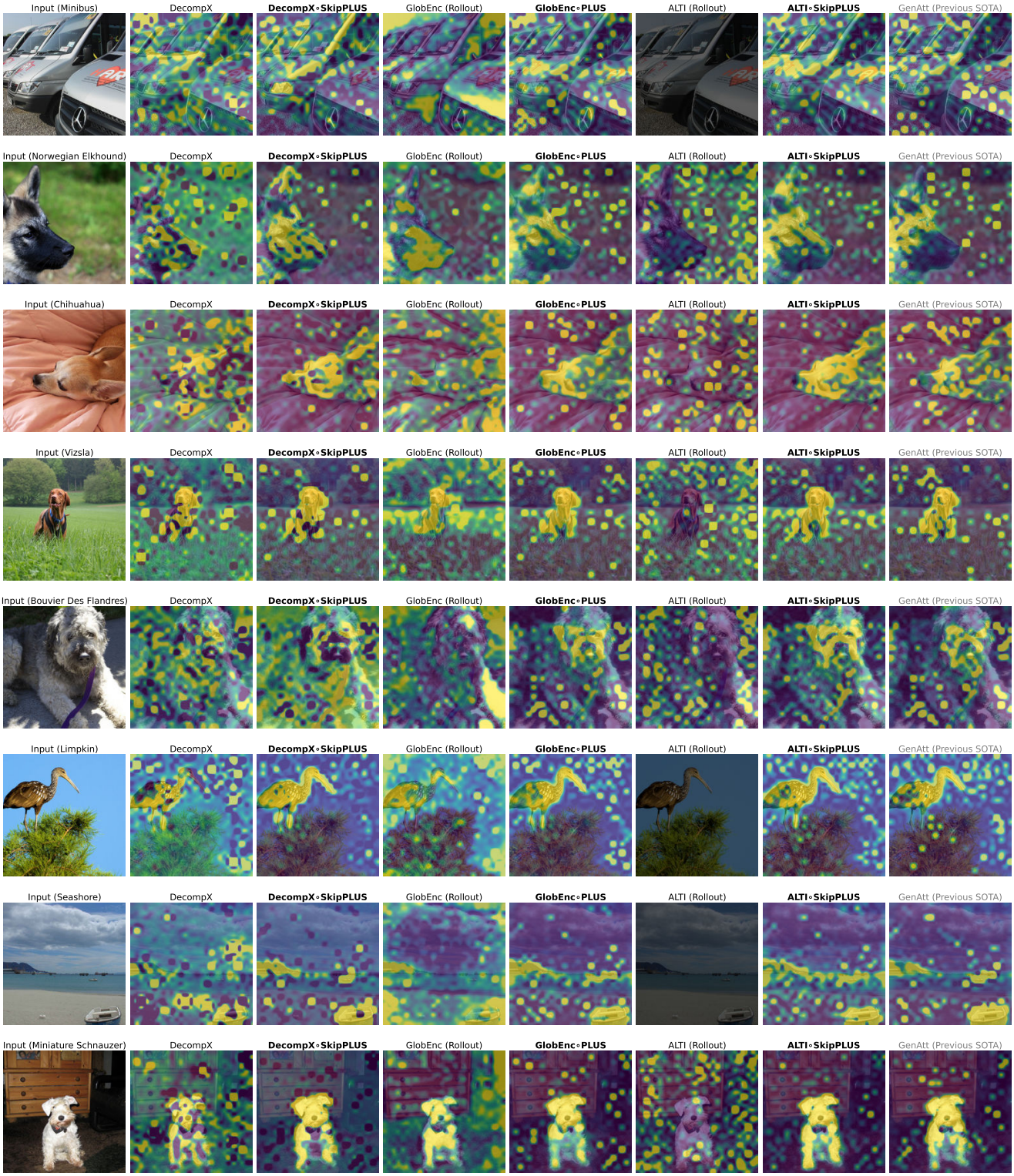

Figure 17. Additional qualitative examples demonstrating the application of SkipPLUS on ViT Base (Patch Size 8). The images presented were selected at random.

D.2. EVA Large (Patch Size 14)

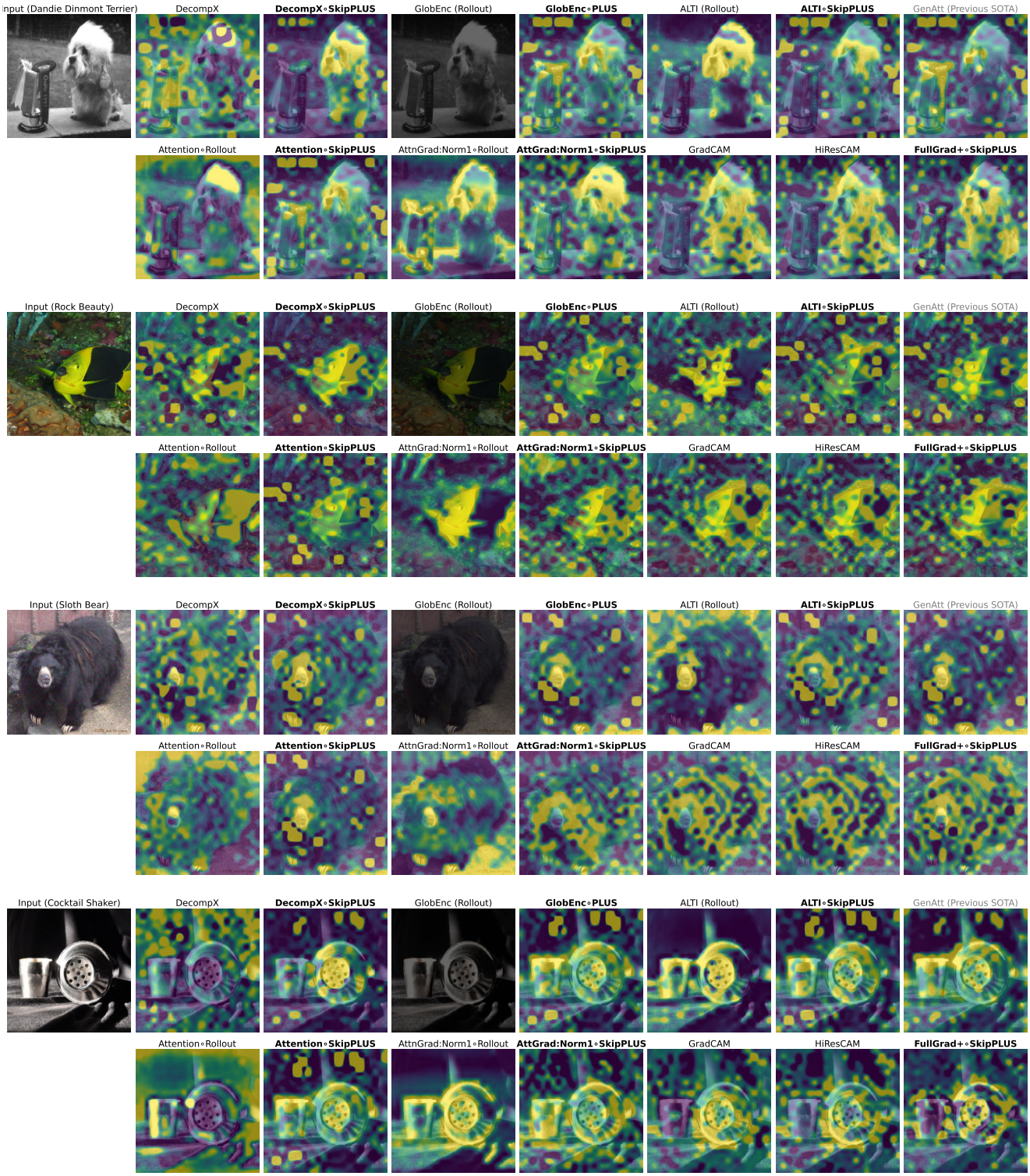

Figure 18. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

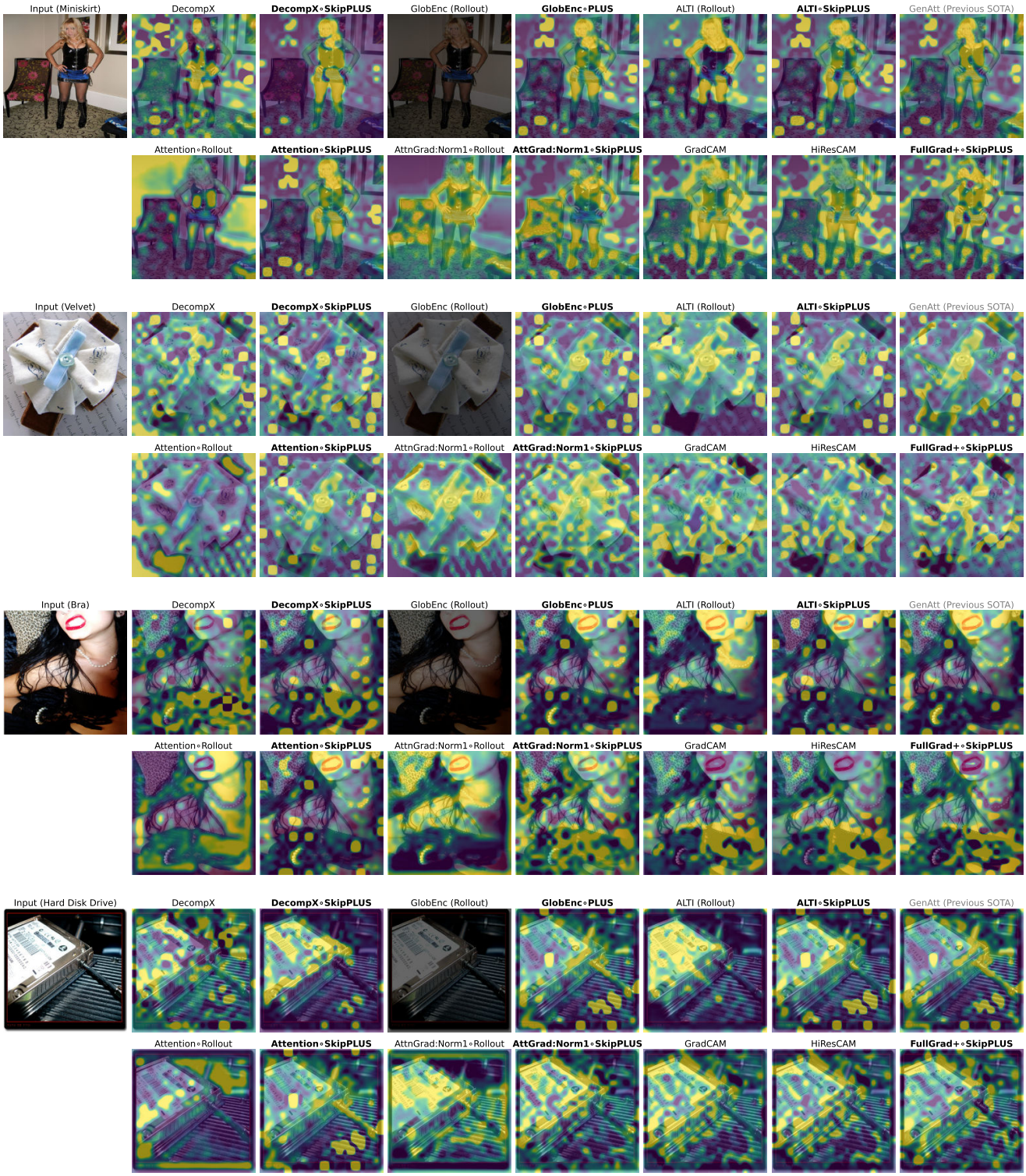

Figure 19. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

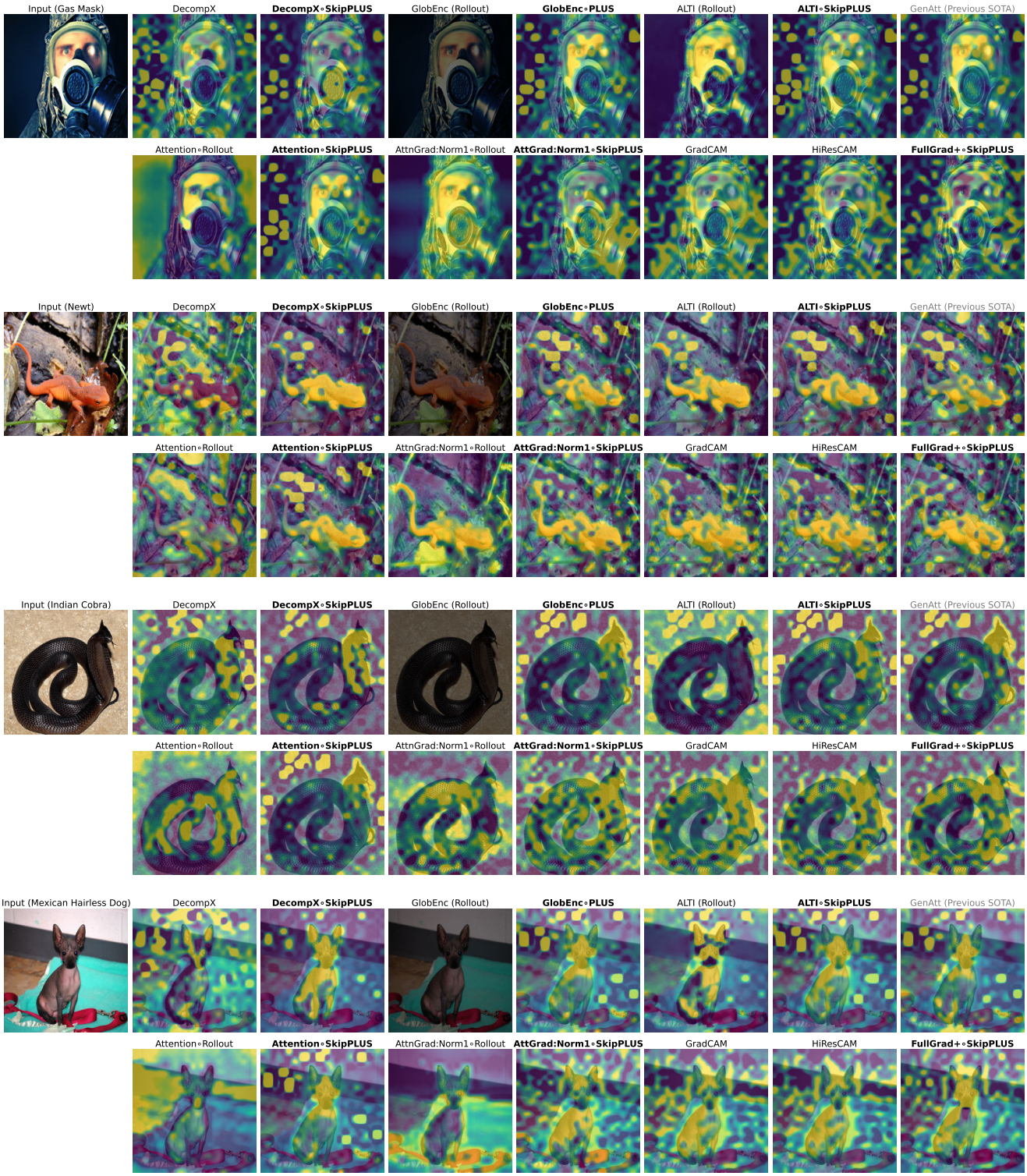

Figure 20. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

Figure 21. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

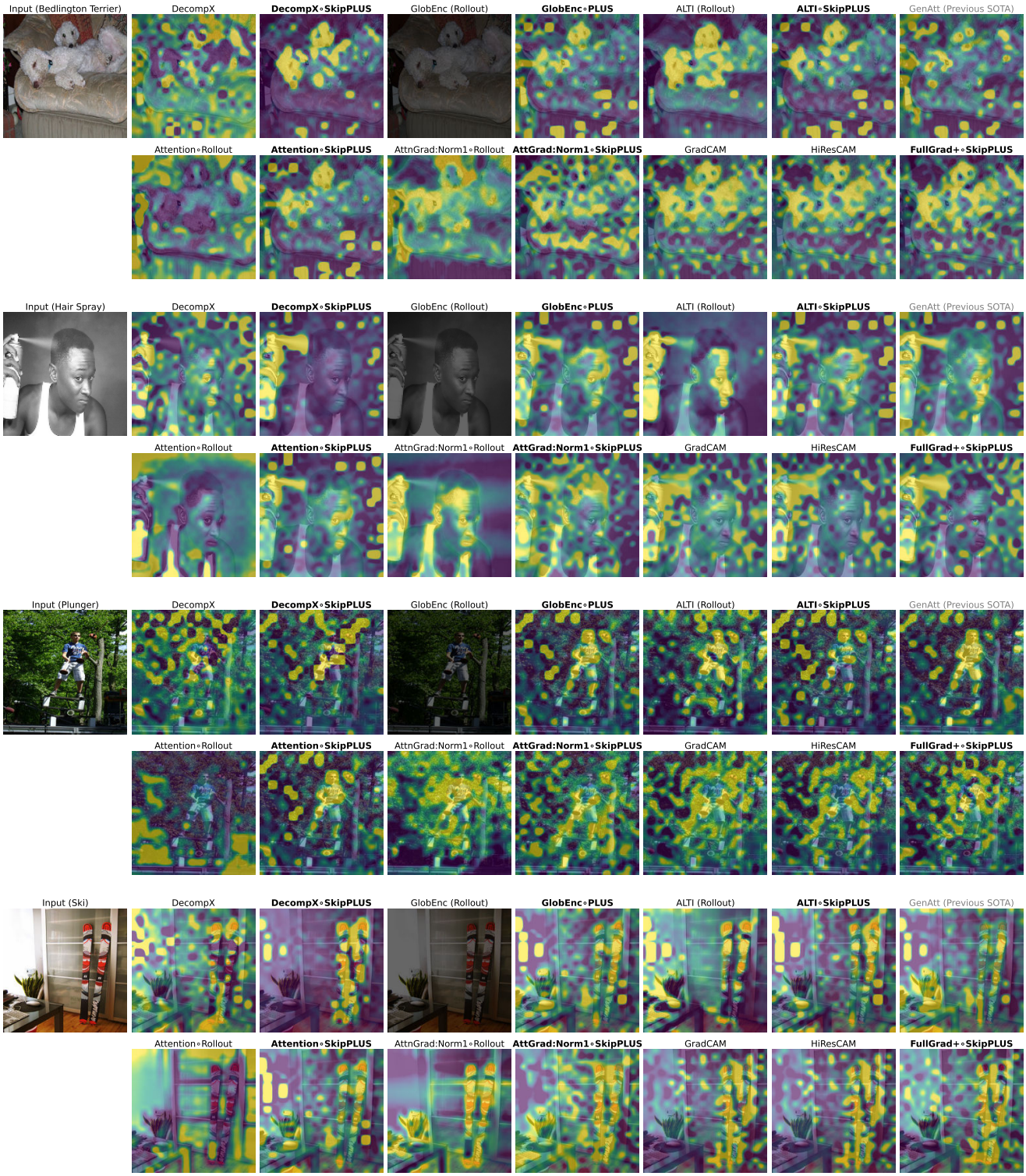

Figure 22. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

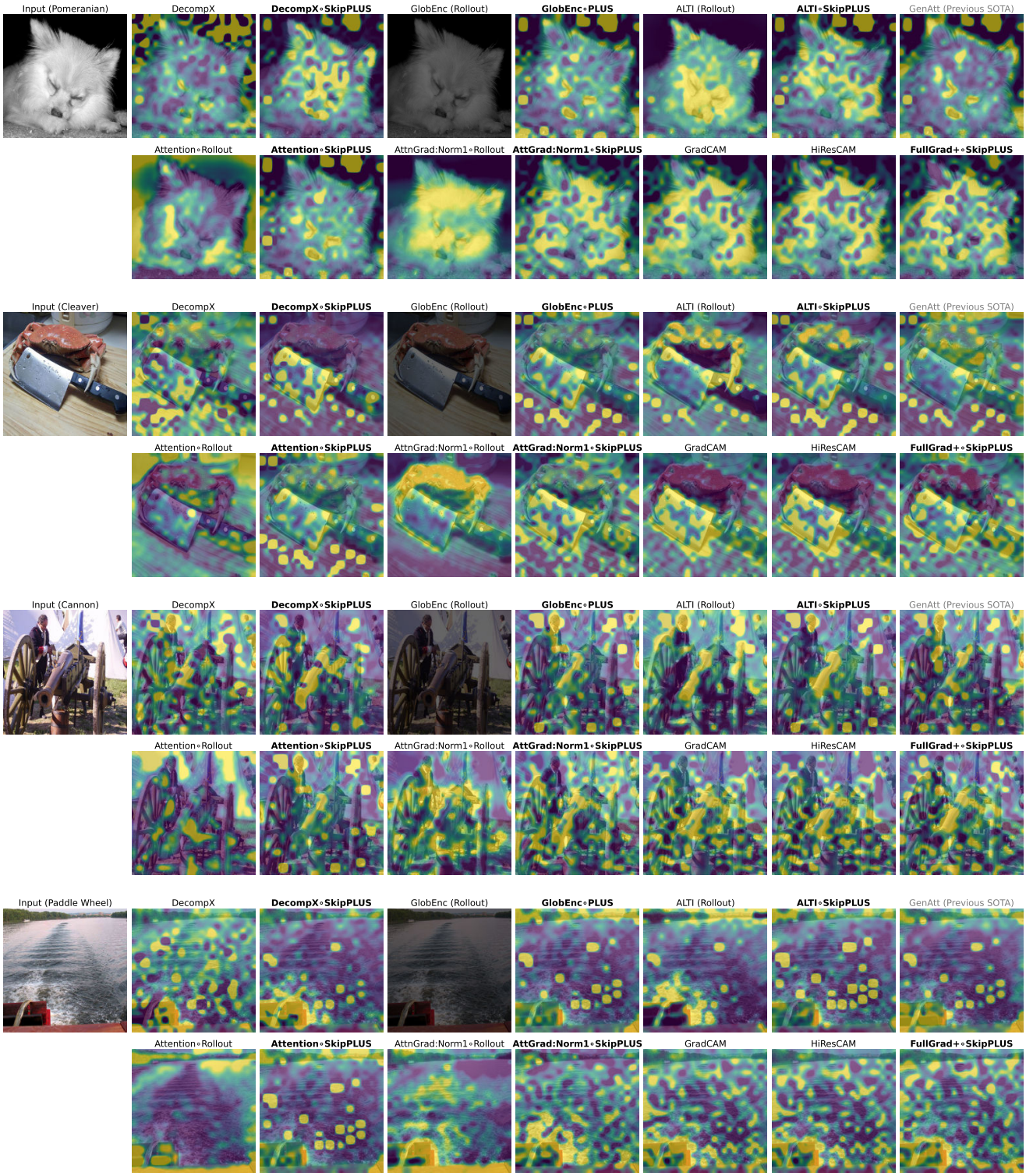

Figure 23. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

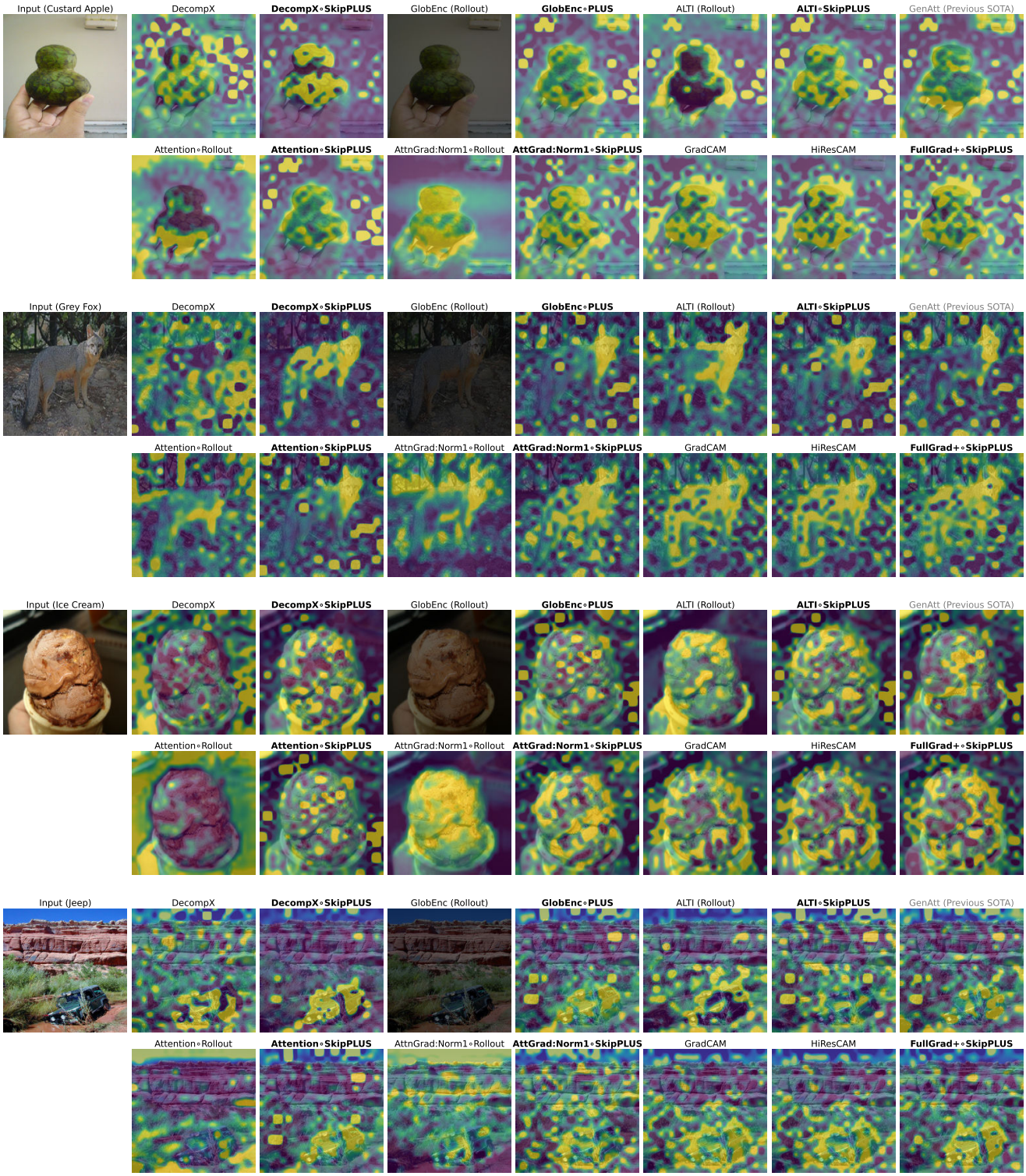

Figure 24. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

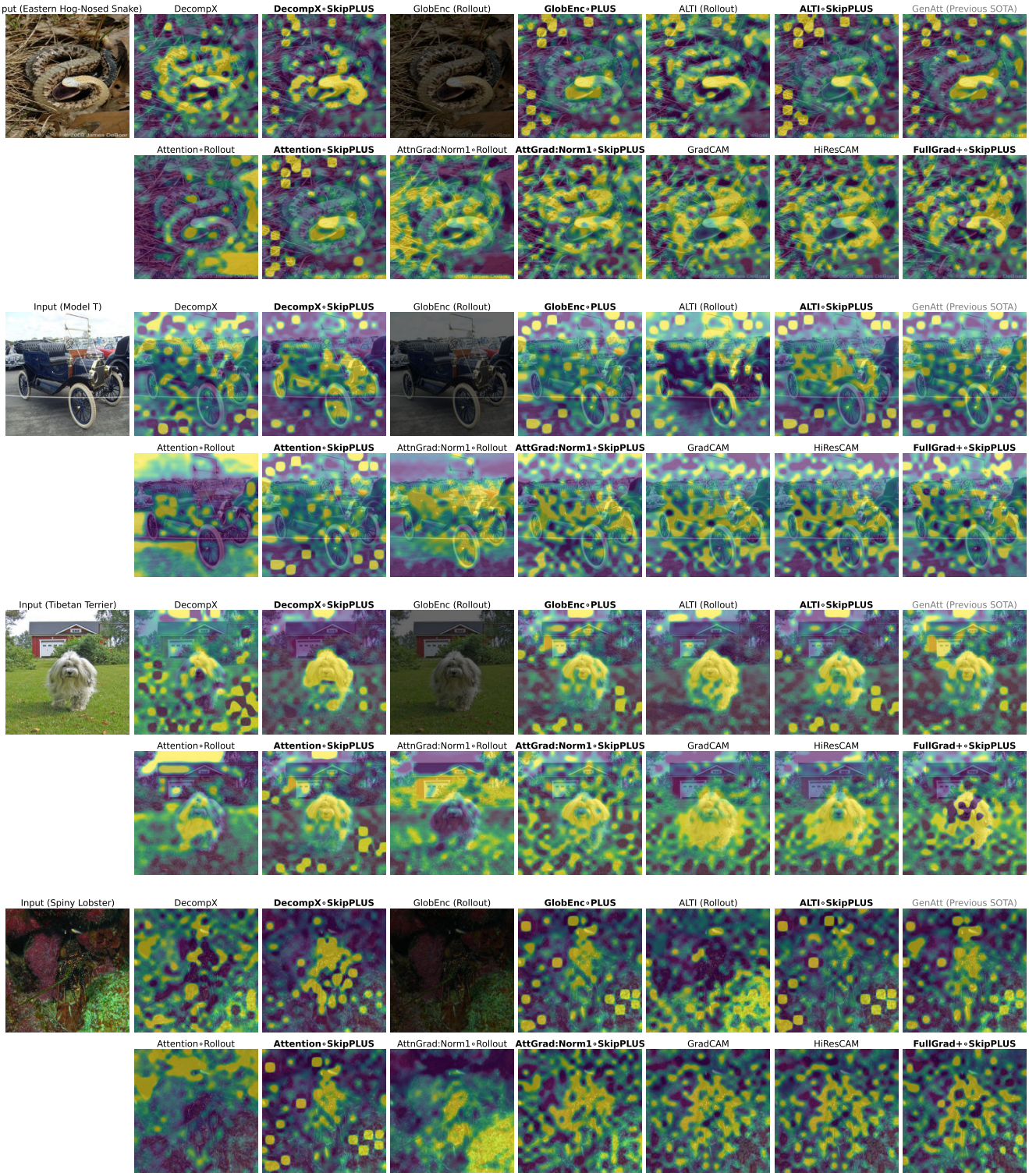

Figure 25. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

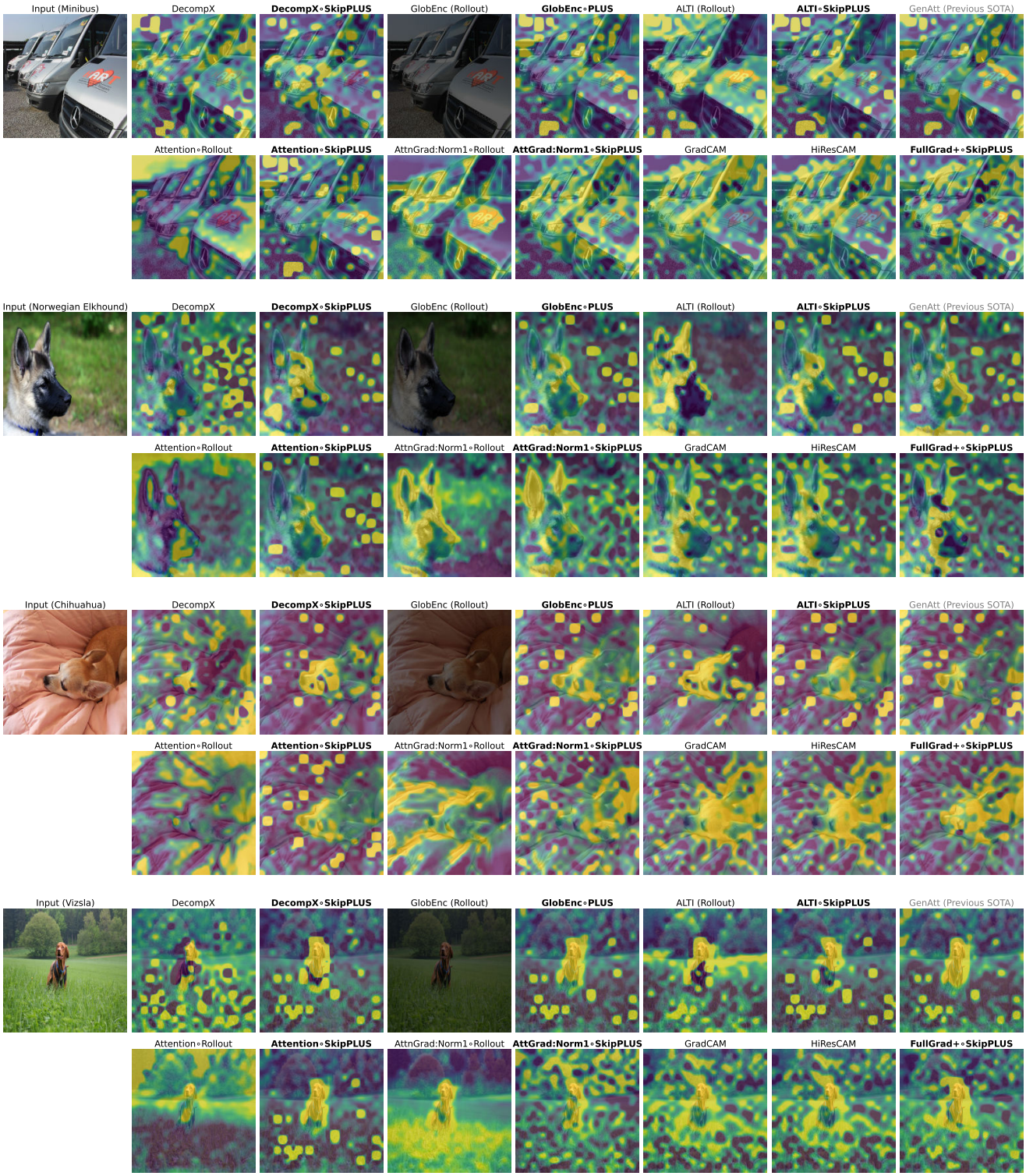

Figure 26. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

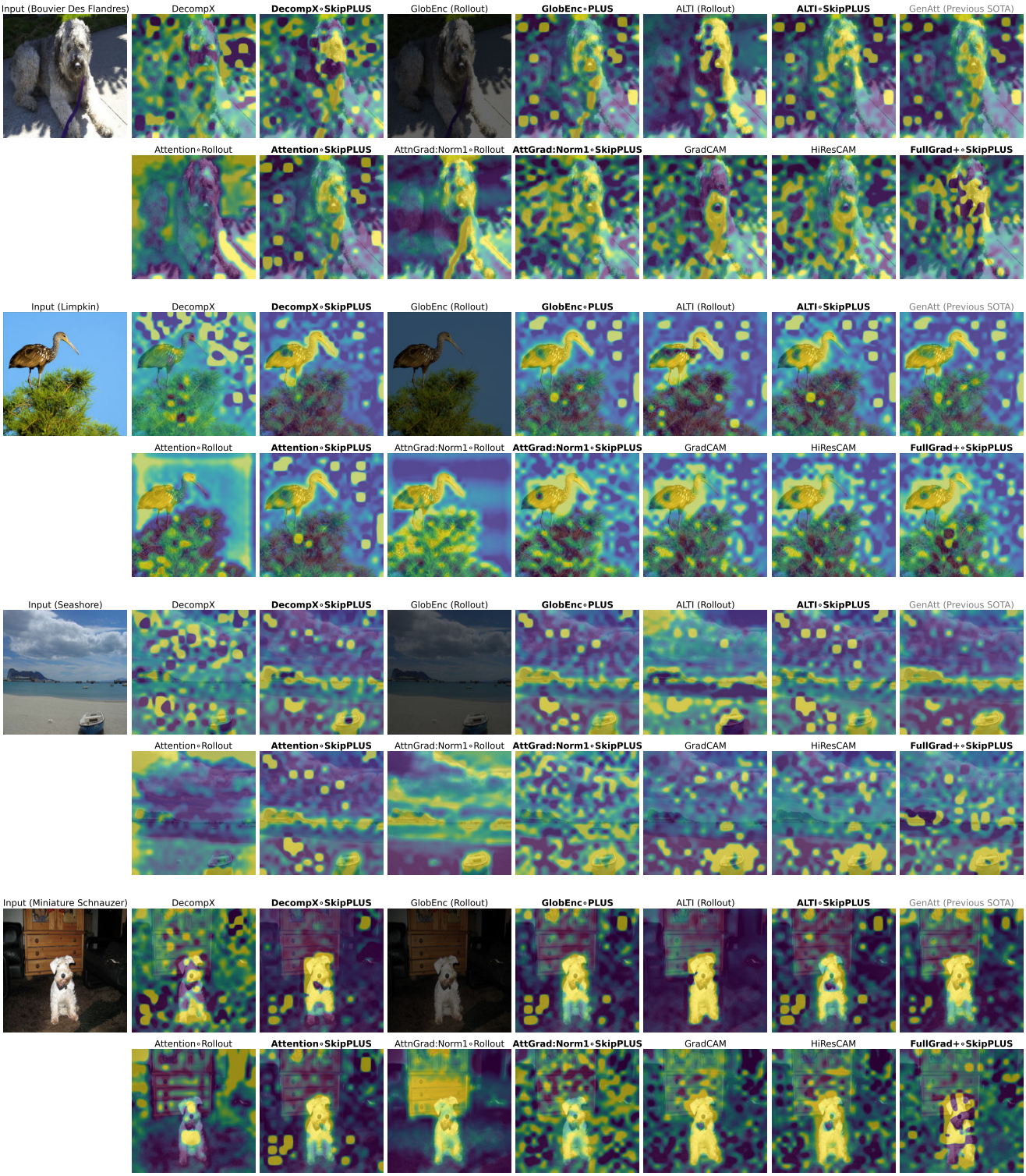

Figure 27. Additional qualitative examples demonstrating the application of SkipPLUS on EVA Large (Patch Size 14). The images presented were selected at random.

E. Related Work

Input attribution methods are techniques designed to quantify the influence of individual input features, or groups of them, on a model's output [5, 37, 42, 43, 60, 65, 66, 77]. Input attribution methods can assist in understanding a model's decision locally for a single input considered in isolation. They also act as foundational elements for more advanced explanation techniques. For instance, in concept-based explanation methods like CRAFT [24], attribution methods are employed for two main purposes: to quantify the impact of each activated concept and to identify the specific input features responsible for activating these concepts.

Attribution methods have a wide array of applications beyond merely explaining model outputs to humans [21, 62, 69, 71]. They are useful for enhancing the robustness of models against out-of-distribution data, spurious correlations, and adversarial inputs [2, 11, 50, 76]. Additionally, attribution methods have been employed to improve the performance of text-to-image models [12, 36, 53]. Furthermore, adapting forward-mode attribution methods has been explored for on-the-fly feature pruning [23, 45] and model quantization [4]. Attribution methods have been utilized to construct more effective adversarial attacks against models [32, 74, 78].

E.1. Gradient-Based Methods

Gradient-based methods compute the gradient of the model's output y_c w.r.t. the input features x_i which can be pixels, regions, or tokens of the whole input x. The general idea is that larger gradients indicate higher importance of the input features x_i on the prediction y_c.

Vanilla Gradients. The most straightforward approach to these methods is to use the gradient as the exact importance score [65].

$$\text{VanillaGrad}_i = \sum_j \frac{\partial y_c}{\partial x_{i,j}}.$$

$$\text{VanillaGrad:Norm2}_i = \left\| \frac{\partial y_c}{\partial x_i} \right\|_2.$$

Input×Gradient (IxG). IxG [37] multiplies the input values by their corresponding gradients. Let x_i be a spatial feature of the input, where $x_{i,j}$ represents the j-th channel of x_i. The Input×Gradient attribution for the spatial feature x_i with respect to the target class c is computed as follows, where y_c is the output of the model for the target class c:

$$\text{Input}\times\text{Gradient}_i = \sum_j \frac{\partial y_c}{\partial x_{i,j}} \cdot x_{i,j}, \tag{7}$$

E.1.1 CAM Methods

CAM methods, popularized by GradCAM [61], usually start from the very last layer of the network. In this, their intuition has certain similarities with SkipPLUS; they both recognize the noisy nature of the first layers of the network.

GradCAM.

- A^k: the k-th channel of the feature map in the final layer
- c: the class w.r.t. which the attribution map is computed
- y^c: the class score (logit)
- Gradients are averaged over the width and height dimensions (indexed by i and j respectively) to obtain the neuron (channel) importance weights α_k^c:

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

AttCAM. The method introduces a Transformer-specific adaptation of GradCAM [61], and reuses the name GradCAM for it [10]. We term this modified method AttCAM.

XGradCAM. It weights the gradients by their corresponding activation value when computing the spatial average [28]. XGradCAM was proposed on ReLU CNNs where the activations were always positive, hence they did not specify using the absolute value of the activations in the above computation, as is more intuitive. We name the variant with absolute activations XGradCAM+, and test both of them. Other CAM methods include GradCAM++ [8], HiResCAM [20], and GradCAMElementWise [30].

E.1.2 Gradient-Based Rollout Methods

TransAtt. TransAtt [10] employs the Deep Taylor Decomposition technique [48] to attribute local relevance and subsequently propagates these relevance scores through the entire architecture of a Transformer model. This process effectively enables the backward propagation of information across all layers, starting from the output and extending back to the input. Additionally, this method incorporates gradients of attention weights. The method's functioning can be summarized as follows:

$$\mathit{Rollout}\left(\mathbb{E}_{H:=\text{Heads}}\left[(\mathtt{R} \odot \mathtt{AttnGrad})^+\right]\right),$$

where R stands for the relevancy scores of attention weights. The Rollout technique is a method to aggregate the layer-wise attribution maps. We refer the reader to [1] for a detailed overview.

GenAtt. The dependence of TransAtt on specific rules for the propagation of relevance scores imposes limitations on its capacity to furnish explanations for various types of Transformer architectures. To cope with this issue, GenAtt [9] attempts to explain predictions for any Transformer-based architecture by using the attention weights in each block to update the relevancy maps, as demonstrated by the following expression:

$$Rollout\left(\mathbb{E}_{H:=\text{Heads}}\left[\left(\texttt{Attn}\odot\texttt{AttnGrad}\right)^{+}\right]\right).$$

The notation $()^{+}$ denotes a filtering through the ReLU function. [9] show that GenAtt is at least as effective as TransAtt, if not better.

E.1.3 Special Cases of PLUS

The methods described below can be viewed as special cases of PLUS, where PLUS is composed with a previously existing method.

GradSAM. GradSAM [3] is equivalent to composing GenAtt [9] with PLUS, instead of using Rollout. (cf. Fig. 9 in the appendices)

CAT. Class Activation Tokens [55] is equivalent to IxG∘PLUS. (cf. Fig. 11 in the appendices)

AttCAT. We can define an attention-enhanced variant of IxG, AttIxG, by multiplying IxG with AttnFrom:

$$\texttt{AttnFrom}_j = \frac{1}{H\times N}\sum_{h=1}^{H:=\text{Heads}}\sum_{i=1}^{N:=\text{Tokens}}\text{RawAttn}_{h,i,j}$$

Note that attention weights have three dimensions: heads, to, from.

Attentive Class Activation Tokens [55, AttCAT] would then be equivalent to AttIxG∘PLUS. (cf. Fig. 11 in the appendices)

LayerCAM. LayerCAM [35] was introduced for ReLU CNN networks, where it is equivalent to applying a normalization process on the layer-wise attribution maps obtained from GradCAMElementWise [30], followed by the PLUS aggregation method. The normalization step is proposed because earlier layers tend to have smaller attribution maps compared to later layers. By normalizing the maps, LayerCAM ensures that each layer contributes more equally to the final attribution map. However, this approach is not suitable for ViTs, as we explicitly want to avoid giving earlier layers the same impact on the final attribution map as later layers (cf. Fig. 2, also supported by our preliminary quantitative evaluations).

E.2. Forward Attention-Based Token Attribution Methods

Although we have mathematically detailed most of the previous methods, the complexity of the subsequent approaches surpasses the scope of this paper. Therefore, we will provide a succinct overview of their core concepts. For a more thorough understanding, we recommend readers refer to the original papers.

Attention×Input_Norm (AttIN). Kobayashi et al. [38] multiply the attention weights by the norms of the vectors corresponding to each attention weight. Kobayashi et al. [39] extends AttIN to also incorporate the residual connections.

GlobEnc & ALTI. AttIN assumes that tokens retain their original identity. As each self-attention module mixes all the tokens, this assumption might not necessarily hold. Using gradient-based techniques, Brunner et al. [6] studies contextual information aggregation across the model. Following Brunner et al. [6] work, the global token attribution analysis method [44, GlobEnc] further extends AttIN by including the Transformer block's second normalization layer in its analysis. In parallel with GlobEnc, the Aggregation of Layer-Wise Token-to-Token Interactions method [26, ALTI] was introduced. ALTI shares core concepts with GlobEnc, but the two differ in certain mathematical specifics.

DecompX. DecompX [46] enhances GlobEnc by integrating the one element previously overlooked by GlobEnc: the MLP module in the Encoder Transformer layer. This inclusion enables DecompX to generate a set of decomposed vectors that collectively sum up to the actual output vector. Unlike GlobEnc and ALTI, which require computing and aggregating layer-wise attribution maps using techniques like Rollout, DecompX facilitates the direct propagation of these decomposed vectors across layers. This capability allows for the direct computation of attribution maps from any layer to any other layer.

In this paper, unless specified otherwise, we utilize the DecompX variation that omits biases, referred to as DecompX W/O Bias in [46]. Our decision stems from our preliminary tests where no significant differences were observed across various methods of handling biases. To sidestep the complexities and hyperparameters introduced by distributing bias attributions among tokens, we opted for the simplest approach. A detailed evaluation of different methods for bias attribution distribution is reserved for future research.

E.3. Black-Box Methods

Black-box attribution methods treat the model as an opaque entity, (partially) disregarding its internal structure and gradients. These methods typically involve perturbing the input and observing the corresponding changes in the model's output to infer the importance of each input feature. However, this approach often comes with significant computational costs due to the need for multiple model evaluations. In contrast, white-box methods leverage the internal structure and gradients of the model, providing a more efficient and fine-grained understanding of the model's behavior.

In this paper, we focus on white-box methods for several reasons. Firstly, they offer a more computationally efficient approach compared to black-box methods. Secondly, and more importantly, black-box methods can be seen as directly optimizing the faithfulness metrics on which we evaluate the attribution methods. This raises concerns related to Goodhart's law, which states that when a measure becomes a target, it ceases to be a good measure. In other words, the faithfulness metrics we use are merely proxies for the ultimate desirable properties we seek in attribution methods. By directly optimizing these metrics, black-box methods may inadvertently introduce biases or artifacts that undermine the true faithfulness of the attributions. Therefore, to avoid this potential pitfall and maintain a more objective evaluation, we refrain from including comparisons with black-box methods in this study, acknowledging that they have different trade-offs and use cases.

LIME [58] explains the predictions of any classifier by learning a local interpretable model around the prediction.

RISE [54] is a black-box approach that generates an importance map indicating the saliency of each pixel for the model's prediction by probing the model with randomly masked versions of the input image and obtaining the corresponding outputs.

PAMI [64] masks the majority of the input and uses the corresponding model output as the relative contribution of the preserved input part to the original model prediction.

ScoreCAM [70] is a post-hoc visual explanation method based on class activation mapping that eliminates the dependence on gradients by obtaining the weight of each activation map through its forward passing score on the target class.

ViT-CX [75] adapts ScoreCAM for ViTs.

AtMan [16] is a perturbation method that manipulates the attention mechanisms of transformers to produce relevance maps for the input with respect to the output prediction.

HSIC [51] is a black-box attribution method based on the Hilbert-Schmidt Independence Criterion, measuring the dependence between regions of an input image and the model's output using kernel embeddings of distributions.