

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 KEYPOINT COUNTING CLASSIFIERS: TURNING VISION TRANSFORMERS INTO SELF- EXPLAINABLE MODELS WITHOUT TRAINING

Anonymous authors

Paper under double-blind review

ABSTRACT

Current approaches for designing self-explainable models (SEMs) require complicated training procedures and specific architectures which makes them impractical. With the advance of general purpose foundation models based on Vision Transformers (ViTs), this impracticability becomes even more problematic. Therefore, new methods are necessary to provide transparency and reliability to ViT-based foundation models. In this work, we present a new method for turning any well-trained ViT-based model into a SEM without retraining, which we call Keypoint Counting Classifiers (KCCs). Recent works have shown that ViTs can automatically identify matching keypoints between images with high precision, and we build on these results to create an easily interpretable decision process that is inherently visualizable in the input. We perform an extensive evaluation which show that KCCs improve the human-machine communication compared to recent baselines. We believe that KCCs constitute an important step towards making ViT-based foundation models more transparent and reliable. Code is available at *anonymised* and in the supplementary.

1 INTRODUCTION

Vision Transformer (ViT)-based (Dosovitskiy et al., 2021) foundation models are becoming increasingly important in computer vision (Han et al., 2023), but are still limited in safety critical domains due to their lack of explainability (Longo et al., 2024). Self-explainable models (SEMs) (Chen et al., 2019; Gautam et al., 2022; Bassan et al., 2025) provide a promising direction within explainable artificial intelligence (XAI) that can address this lack of explainability. SEMs are a type of algorithm where the decision process is inherently explainable, which is highly important to address the disagreement problem (Krishna et al., 2024) that standard post-hoc explainability methods suffer from. However, SEMs have several limitations that limit their usability (Hoffmann et al., 2021), and here we highlight two key limitations:

(1) SEMs lack flexibility A highly contributing factor to the usefulness of ViT-based foundation models is their high flexibility and generalization capabilities. Often, they can be applied directly to a new tasks without the need for finetuning or adjustments. However, SEMs regularly assume a particular architecture, for example a convolutional neural network (CNN)-based feature extractor (Chen et al., 2019; Nauta et al., 2023), which makes them incompatible with ViT-based foundation models. Moreover, SEMs that do not have these constraints, often require an additional classification head to be trained on top of the ViT (Turbé et al., 2024). This requires additional training, which reduces the flexibility of the entire systems. Therefore, there is a need for new methodology that can turn ViT-based foundation models into SEMs while keeping their flexibility.

(2) Current SEMs visualize explanations poorly In computer vision, the explanations of SEMs are either presented as bounding boxes (Nauta et al., 2023) or heatmaps (Gautam et al., 2024). These two approaches are both standard methods in SEMs, despite the fact that many studies have pointed out limitations associated with both bounding boxes and heatmaps. For bounding boxes, the construction of the bounding boxes entails finding a rectangle that contains a certain amount of relevance. Several works have pointed out that these bounding boxes can be misleading and cover

more area than the actual activation site (Hoffmann et al., 2021; Davoodi et al., 2023). Heatmaps are usually overlaid on an image to highlight what region are important for a decision, and have been criticized for lacking precision (Carmichael et al., 2024), being uninformative (Xu-Darme et al., 2023), and having low performance in user studies (Kim et al., 2023). To improve the usability of SEMs, new methods for visualizing explanations must be developed.



Figure 1: A demonstration of KCCs in the context of bird classification. KCCs identify matching keypoints between a query (leftmost image) and a set of prototypes (rightmost images). Only prototypes with matches are shown to avoid overloading the reader. Predictions are made by counting the number of matches. Here, we leverage ViTs with vision-language capabilities to automatically describe the keypoints (Sun et al., 2023). Note that the class names in the predictions are deliberately omitted to avoid readers using the class names instead of the explanation Kim et al. (2022).

In this work, we address these limitation through a novel framework called Keypoint Counting Classifiers (KCCs), which can turn any ViT-based foundation model into a SEM without the need to train the feature extractor or an additional classification head. KCCs compare *parts* of the query image with *parts* from the prototypes, which is accomplished by exploiting the part-correspondence in the tokens of ViTs (Amir et al., 2022). A set of matching regions between the query and the prototypes is identified through the use of mutual NNs (Oron et al., 2018), and the final classification decision is reached by counting matches between the query and the prototypes. This procedure is performed without any retraining, thus addressing the lack of flexibility. Then, the explanations are visualized as matching keypoints, which is a completely new approach for visualizing explanation that aims to improve the communication of the explanations.

The use of keypoints to visualize explanations is motivated by its frequent use in teaching material and visual learning. One example is in the study of birds, where keypoints are commonly used to visualize critical parts of the animal, which even has its own name (bird topography (Pettingill, 1984)). Another example is in the study of human anatomy, where keypoints are the standard way to highlight important anatomical features in teaching material (see for example (Netter, 2017)). Studies on how people learn most effectively have highlighted that people can only process a few pieces of information at the time to reduce cognitive overload (Mayer, 2018). Our motivation for introducing keypoints is to avoid this overload that uninformative and imprecise heatmaps and bounding boxes could introduce. Additionally, we show how in the special case of ViTs with vision-language capabilities, the keypoints can be automatically labeled, which is motivated from the perspective of reducing reader bias Bove et al. (2024). Figure 1 show two examples if the explanation provided by a KCC in the conext of bird classification. In summary, KCCs constitute a new paradigm among SEMs with increased flexibility and improved visualization of explanations. Our contributions are:

1. We introduce Keypoint Counting Classifiers; a general purpose method for turning ViT-based foundation models into SEMs without any retraining.
2. We conduct an extensive quantitative evaluation with comparison to recent and relevant baselines that demonstrate the benefits of KCCs.
3. We conduct a human user study which shows that KCCs improve human-machine communication compared to existing alternatives.
4. We show how in the particular case of ViTs with vision-language capabilities, keypoints can be automatically labeled which makes initial progress towards reudcing reader bias.

108 **2 RELATED WORK**

110 Creating self-explainable models through prototypical parts is one of the main directions within
 111 self-explainability, and is often motivated from the perspective of recognition-of-component the-
 112 ory (Biederman, 1987). The pioneering work of ProtoPNet (Chen et al., 2019) and its numerous
 113 derivatives (Donnelly et al., 2022; Rymarczyk et al., 2022; Nauta et al., 2023) paved the way for
 114 self-explainable deep learning-based approaches for image classification. The core idea of these ap-
 115 proaches is to use a CNN-based encoder that retains some spatial resolution and learn interpretable
 116 part detectors that are linearly combined to perform classification. A fundamental limitation of these
 117 approaches is that they are constrained to particular CNN-based architectures and can therefore not
 118 take advantage of advances in ViT-based foundation models. Recent works have proposed self-
 119 explainable models built specifically on ViTs. The ViT-NeT combines a ViT backbone with a tree-
 120 based classifier (Kim et al.), and the ProtoS-ViT combined a ViT backbone with additional trainable
 121 layers and a compactness regularization (Turbé et al., 2024) . However, both of these approaches
 122 require an additional training step that limits their flexibility. Some notable works can turn pre-
 123 trained models into self-explainable models. The TesNet provides a plug-in transparent embedding
 124 space (Wang et al., 2021), but is restricted to work on CNNs. More recently, the KMEx framework
 125 can turn any pretrained model into a SEM (Gautam et al., 2024). This is accomplished by creating a
 126 1-nearest-neighbor (NN) classifier in combination with feature visualization techniques. However,
 127 the 1-NN setup restricts the explanation, as only a single image can be visualized as the basis of the
 128 explanation. Thus, there is a need for new training-free methodology with greater flexibility.

129 **3 KEYPOINT COUNTING CLASSIFIER - TURNING ViT-BASED FOUNDATION
 130 MODELS INTO SEMS**

132 We present a new paradigm for SEMs, a keypoint-based approach that requires no additional train-
 133 ing and a new way of visualizing explanations. Our motivation for introducing KCCs is the high
 134 performance and flexibility of ViT-based foundation models in classification and keypoint matching
 135 without additional training (Oquab et al., 2024; Amir et al., 2022) and the frequent use of keypoints
 136 in teaching and visual learning (Pettingill, 1984; Netter, 2017). KCCs are constructed through three
 137 parts: (1) an image-wise identification of keypoints, (2) keypoint matching through mutual NNs,
 138 and (3) classification through counting matching keypoints. These steps are described below.

140 **Preliminaries on ViTs** Let $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ denote an image with C channels, a width of W pixels
 141 and a height of H pixels that is reshaped into a sequence of flattened 2D patches $\mathbf{X}^{(p)} \in \mathbb{R}^{N_T \times P^2}$,
 142 where $N_T = HW/P$ and P is the height and width of each image patch. A ViT transforms $\mathbf{X}^{(p)}$ into
 143 a set of latent representations, referred to as tokens, $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_{N_T}\}$ and a classification token
 144 \mathbf{z}_{cls} that summarizes the information across all tokens into a single representation. All tokens are
 145 of the same dimensionality D . A positional embedding is added to each patch embedding to retain
 146 positional information from the input space. Due to the positional embeddings, the set of tokens can
 147 also be rearranged into a low resolution encoded representation of the input $\mathbf{Z} \in \mathbb{R}^{\sqrt{N_T} \times \sqrt{N_T} \times D}$.

148 **3.1 PART 1 - IMAGE-WISE IDENTIFICATION OF KEYPOINTS**

151 The fundamental idea of KCCs is to count matching keypoints between a query and a set of proto-
 152 types. The first part of KCCs is therefore to identify keypoints in the query and all prototypes. A
 153 recent work by Amir et al. (2022) showed that the tokens of ViT capture information about semantic
 154 parts, and we will build on these findings to identify keypoints.

155 First, we identify foreground pixels in \mathbf{X} . This is necessary since tokens also can capture infor-
 156 mation about the background, which should not be included when matching keypoints between
 157 objects. Formally, let f be a function that takes in an image \mathbf{X} and outputs a binary foreground
 158 mask $\mathbf{F} \in \{0, 1\}^{H \times W}$, where 1 indicates foreground pixels and 0 indicates background pixels.
 159 Foreground identification is a fundamental problem in computer vision that can be handled in many
 160 ways, and here we treat the selection of f as a task-specific hyperparameter. In some cases, privi-
 161 liged information in the form of segmentation masks could be available. When privileged infor-
 162 mation is not available, general purpose foundation models have been shown to provide excellent

foreground segmentation across a diverse set of visual tasks (Ren et al., 2024). If external models are not accessible, the ViT itself can be used to identify foreground pixels through the attention mechanism (Amir et al., 2022) or through subspace analysis of the tokens (Oquab et al., 2024). In this work, we rely on the combination of the Segment Anything model (Kirillov et al., 2023) and Grounding DINO to perform the foreground segmentation, due to its impressive performance across numerous diverse and challenging tasks (Ren et al., 2024).

Second, the foreground is split into semantically coherent segments. These segments will form the basis for the keypoints and is computed based on the tokens due to their strong alignment with object parts (Amir et al., 2022) through the \mathbf{Z} representation of the image \mathbf{X} . A challenge when operating on \mathbf{Z} is that the resolution is much lower than the input and not compatible with the foreground mask \mathbf{F} . We found that a simple solution to this challenge was to downsample \mathbf{F} using NN interpolation and upsample \mathbf{Z} using bilinear interpolation to a new resolution $H' \times W'$ between $H \times W$ and $\sqrt{N_T} \times \sqrt{N_T}$, was a good compromise between reducing computational demand and avoiding overly coarse part-segments.

Mathematically, let s be a function that takes in the upsampled token representations \mathbf{Z} and outputs a discrete mask $\mathbf{S} \in \{0, \dots, N_s\}^{H' \times W'}$, where each integer indicates membership to a semantically coherent segment and N_s is the number of segments for that particular image. Again, we treat this process as a task-specific hyperparameter. If available, privileged information about part locations can be used, and if not, general purpose methods based on traditional image processing techniques or more recent deep learning-based approaches are available (Achanta et al., 2012; Aniraj et al., 2024). Compared to the foreground identification task, part-segmentation is more challenging and current deep learning approaches often must be trained for a particular dataset (Aniraj et al., 2024). Therefore, to retain flexibility and avoid additional training we leverage the widely used SLIC method (Achanta et al., 2012) to separate the foreground into semantically coherent segments due to its reliability and flexibility.

Finally, we define a keypoint as the center of a segment. The entire first part of KCCs is illustrated in columns 1-4 of Figure 2. Since this part of the method is performed image-wise, the entire procedure can be precomputed for the prototype keypoints, which can greatly speed up inference in cases where many prototypes are necessary.

3.2 PART 2 - IDENTIFY MATCHING KEYPOINTS

Once the keypoints have been located, the second part of KCCs is to identify matching keypoints. To perform the matching, we take inspiration from template matching using mutual NNs (Oron et al., 2018), which has been demonstrated to effectively locate matching keypoints between two images (Amir et al., 2022).

The mutual NN procedure requires computing similarities between the keypoints. Here, we propose to represent each keypoint as the mean representation of all tokens within a segment of \mathbf{S} , and compute similarities between these representations. Formally, we calculate the mean representation of segment i for a given image \mathbf{X} as:

$$\boldsymbol{\mu}_i = \frac{1}{K_i} \sum Z(j), \text{ for all } j \text{ where } S(j) = i, \quad (1)$$

where K_i is the number of tokens belonging to segment i . Next, we can determine if two segments are mutual NNs as follows:

$$MNN(\boldsymbol{\mu}_i^q, \boldsymbol{\mu}_j^{p_k}, U^q, U^P) = \begin{cases} 1, & \text{if } NN(\boldsymbol{\mu}_i^q, U^P) = \boldsymbol{\mu}_j^{p_k} \text{ and } NN(\boldsymbol{\mu}_j^{p_k}, U^q) = \boldsymbol{\mu}_i^q \\ 0, & \text{else} \end{cases}. \quad (2)$$

where

$$NN(\boldsymbol{\mu}_i^q, U^P) = \arg \min_{\boldsymbol{\mu}^P \in U^P} d(\boldsymbol{\mu}_i^q, \boldsymbol{\mu}^P) \quad (3)$$

and $d(\cdot, \cdot)$ is some distance measure. In Equation 2, $\boldsymbol{\mu}_i^q$ is the representation of segment i in the query, $\boldsymbol{\mu}_j^{p_k}$ is the representation of segment j in prototype k , U^q is the set of all query segment representations, and U^P is the set of all k prototype segment representations. Equation 2 is repeated

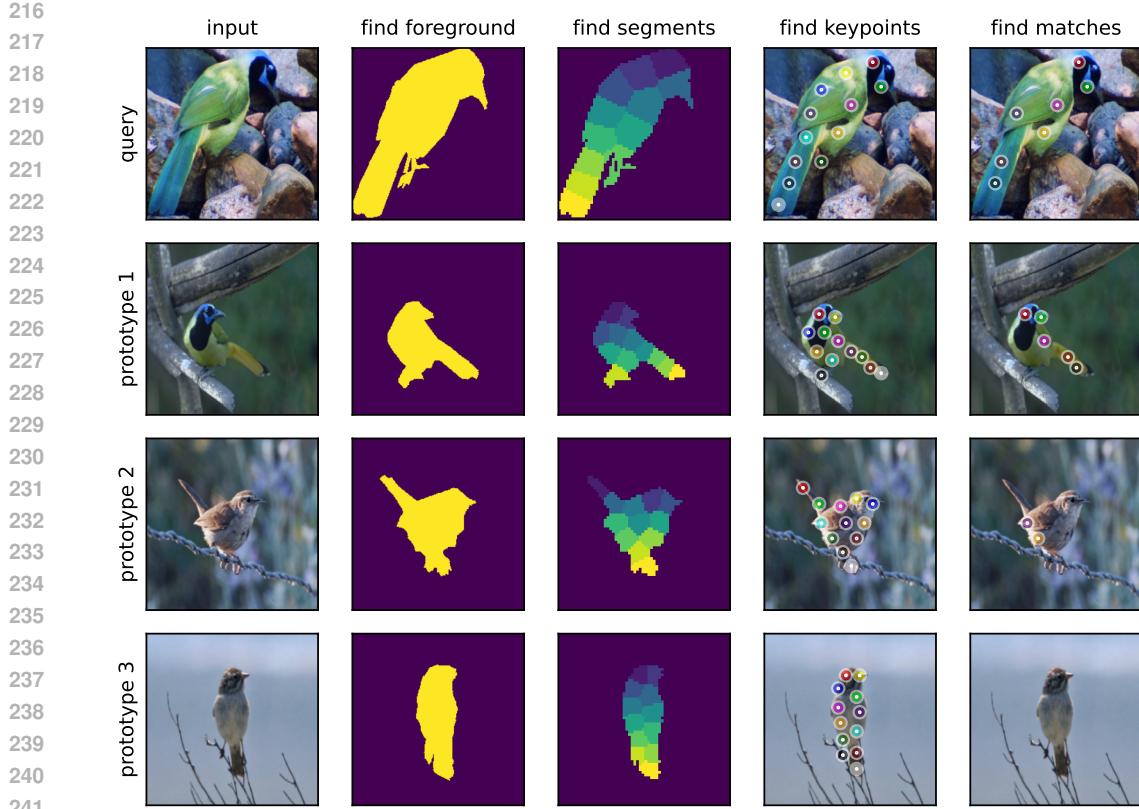


Figure 2: Illustration of each step of KCCs. Going from column 4 to column 5, mutual NNs are computed between the keypoints in the query and all prototype keypoints. Only keypoints that are mutual NNs kept. In this case, prototype 3 has no mutual NNs with the query, and is therefore without keypoints.

across all query and prototype tokens, and tokens that are identified as mutual NNs are collected in the set M . Additionally, we create a corresponding set Y^M that contains the class label of the matched prototype. The last column Figure 2 shows the remaining keypoints after the matching.

3.3 PART 3 - CLASSIFY BY COUNTING MATCHING KEYPOINTS

After identifying matching keypoints, the final part of KCCs is to classify the query based on the number of matches. For a given class c , we count the number of matches between the query and prototypes belonging to the image-wise class c as:

$$\hat{y}_c = \frac{1}{|M|} \sum Y^M(i), \text{ for all } i \text{ where } Y^M = c. \quad (4)$$

4 EVALUATION AND EXPERIMENTAL SETUP

We describe how KCCs are evaluated, how we identify prototypes, and how we select and evaluate important hyperparameters in KCCs.

Evaluation We follow the established evaluation protocol for SEMs (Chen et al., 2019; Nauta et al., 2023; Rymarczyk et al., 2022; Turbé et al., 2024) and measure accuracy and complexity. Complexity is measured by calculating the average number of images a user must inspect as part of the explanation. We evaluate on widely used fine-grained classification datasets; CUB200 (Wah et al., 2011), Stanford Cars (Krause et al., 2013), and Oxford Pets (Parkhi et al., 2012), and com-

pare with numerous strong baselines; ProtoPNet(Chen et al., 2019), ProtoTree (Nauta et al., 2021), ProtoPool (Rymarczyk et al., 2022), PIP-Net (Nauta et al., 2023), ProtoS-ViT (Turbé et al., 2024), ViT-NeT (Kim et al.), ST-ProtoPNet (Wang et al., 2023), and KMEx (Gautam et al., 2024).

User study We conducted a comprehensive user study to evaluate how users perceive different explanation types generated by SEMs. The study design was based on the HIVE framework, which allows for falsifiable hypothesis testing, cross-method comparison, and human-centered evaluation (Kim et al., 2022). In particular, we included the agreement task and subjective evaluation questions, and excluded any biasing factors like class labels. The study compared three visualization methods: bounding boxes, heatmaps, and our proposed keypoints. Drawing on prior research indicating user preference for example-based and part-based explanations (Kim et al., 2023), we included two baseline methods: PiP-Net (Nauta et al., 2023), which offers prototype-based explanations, and KMEx (Gautam et al., 2024), which uses NN examples (i.e. example-based). PiP-Net explanations were visualized using bounding boxes, whereas KMEx employed heatmaps. Each method was represented by six correct and six incorrect classifications, with participants viewing three randomly selected examples per method. All examples were drawn from the same set of classes across methods to control for class difficulty. The order of methods was randomized between participants. Before the study, all participants received a short lecture explaining all methods to ensure a common understanding of the explanations. We evaluate the *agreement* with the model prediction, the *quality* of the shown explanation and the *understanding* of the explanation as follows:

- *agreement*: How confident are you that the model prediction is correct based on the explanation? (scale 1-4, 1/2: prediction is incorrect, 3/4: prediction is correct)
- *quality*: Do you agree with the key point matches/heatmap/matched patches? (scale 1-4, 1/2: keypoints/heatmap/patches are incorrect, 3/4: keypoints/heatmap/patches are correct)
- *understanding*: How easy is it to understand the explanation? (Scale 1-4, 1: Very difficult, 4: Very easy)

For *agreement* we report how many identified the correct and incorrect predictions correctly. For *quality* and *understanding*, we report the mean answer and standard deviation. Differences in methods are compared with a 2-sample t-test (p-value < 0.05 is considered significant).

Overall, 51 participants with medium to high expertise in AI (3.66 ± 0.66 , scale 1-5 (1:no experience, 5:expert)) and low expertise in birds (1.51 ± 0.75 , scale 1-5 (1:no experience, 5:expert)) participated in the study.

Finding prototypes and hyperparameters Following prior works Chen et al. (2019); Gautam et al. (2022), we calculate 10 prototypes per class, and we use the cosine similarity in Equation 3. Note, since similarity is computed between all tokens in the query and all tokens in all prototypes, this can quickly become memory intensive if there are many classes. Therefore, we only consider the distance to tokens in the query to tokens of the J closest prototypes (in terms of cosine similarity). We evaluate the performance of KCCs for several choices of J in the experiments. For the number of segments (N_s), we report results for 8 and 12 segments per image.

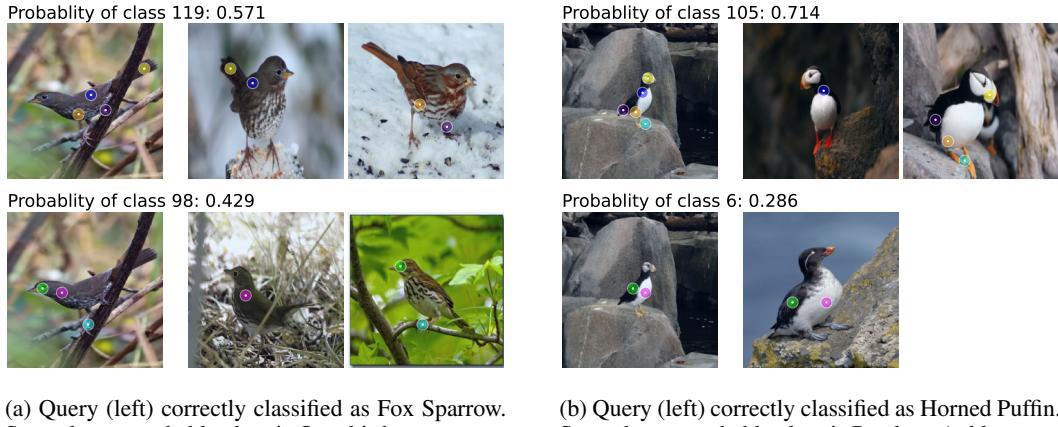
5 RESULTS

We present the main results of our evaluation. First, we show the results of the user study, followed by the performance evaluation in terms of accuracy and complexity. Then, we show how in the particular case of ViT-based foundation models with vision-language capabilities, KCCs can be imbued with automatically generated keypoint descriptors in the form of text to reduce reader bias and enhance user friendliness. In Appendix B, we perform a study of hyperparameters like the number of keypoints and prototypes and evaluate the performance across different ViTs.

User study We conducted a user study to evaluate the three explanation methods with different visualizations: PIP-Net, KMEx, and our proposed method, KCC. The study assessed participants' ability to identify correct and incorrect predictions (*agreement*), perceived explanation *quality*, ease of *understanding*, and overall *user preference*, the main results are shown in Table 1.

324
 325
 326
 327
 328
 329
 Table 1: Results of user study. The table reports user agreement with model predictions (correct
 and incorrect), perceived explanation quality (mean \pm std), ease of understanding, and overall user
 preference. Bold numbers indicate statistical significance (2-sample t-test, p-value < 0.05). KCC
 achieved the highest scores in explanation quality and understanding, and was tied with KMEx as
 the most preferred method. The agreement task highlights the confirmation bias (Skitka et al., 1999);
 users trust the model’s prediction even if it is wrong.

331 method	332 type of visualization	333 prediction	334 agreement	335 quality	336 understanding	337 user preference
333 PIP-Net	334 bounding box	335 correct	336 77.92%	337 3.06 ± 0.68	338 2.53 ± 0.77	339 17.6%
		340 incorrect	341 46.05%	342 2.70 ± 0.67	343 2.51 ± 0.79	
344 KMEx	345 heatmap	346 correct	347 85.71%	348 2.84 ± 1.00	349 2.99 ± 0.90	350 41.2%
		351 incorrect	352 35.53%	353 2.50 ± 1.15	354 2.78 ± 0.92	
355 KCC (ours)	356 keypoint	357 correct	358 88.16%	359 3.34 ± 0.62	360 3.05 ± 0.71	361 41.2%
		362 incorrect	363 35.06%	364 3.10 ± 0.82	365 2.81 ± 0.80	



353
 (a) Query (left) correctly classified as Fox Sparrow.
 354 Second most probable class is Ovenbird.

355
 (b) Query (left) correctly classified as Horned Puffin.
 356 Second most probable class is Parakeet Auklet.

357
 358
 Figure 3: Qualitative examples of KCC explanations.

359
 360
 361
 362
 363
 364
 365
 366
 KCC achieved the highest rate of correct prediction identification (88.16%), significantly outper-
 367 forming PIP-Net (77.92%, $p < 0.05$). KMEx also performed well (85.71%), the difference compared
 368 to KCC was not statistically significant. For incorrect predictions, PIP-Net had the highest agree-
 369 ment rate (46.05%), but this difference was not statistically significant compared to the other meth-
 370 ods. Across all methods, users correctly identified only 35-46% of incorrect predictions, indicating
 371 a high degree of confirmation bias. This aligns with previous observations in the HIVE study (Kim
 372 et al., 2022), suggesting that users tend to trust model outputs even when they are incorrect when
 373 they are shown explanations.

374
 375
 376
 377
 Participants rated KCCs highest in explanation quality (3.34 ± 0.62 for correct predictions), signif-
 378 icantly outperforming both KMEx (2.84 ± 1.00) and PIP-Net (3.06 ± 0.68). For incorrect predic-
 379 tions, KCCs again received the highest quality ratings (3.10 ± 0.82), which means that the keypoint
 380 matches are mostly perceived as correct, explaining the high trust in the models prediction even if it
 381 is wrong. The quality of the heatmaps for KMEx was generally perceived as moderate (2.50 ± 1.15),
 382 indicating some ambiguity in interpretation. However, this did not appear to reduce user confidence
 383 in the models predictions, potentially due to the intuitive nature of NN examples. In terms of ease
 384 of understanding, KCC was rated the most comprehensible (3.05 ± 0.71), significantly better than
 385 PIP-Net (2.53 ± 0.77), but not significantly different from KMEx (2.99 ± 0.90).

386
 387
 388
 When asked to indicate their preferred method, users favored KMEx and KCC equally (41.2% each),
 389 with PIP-Net receiving the fewest votes (17.6%). This suggests that while KCC offers the best
 390 overall interpretability, KMEx remains a strong alternative.

method	ViT?	no retraining of encoder	no training of clf head	CUB200		CARS		PETS	
				A	C	A	C	A	C
ProtoPNet	✗	✗	✗	79.2	2000	86.1	1960	-	-
ProtoTree	✗	✗	✗	82.2	8.3	86.6	8.5	-	-
ProtoPool	✗	✗	✗	85.5	202	88.9	195	-	-
PIP-Net	✗	✗	✗	84.3	4.0	88.2	4.0	92.0	2.0
ProtoS-ViT	✓	✓	✗	85.2	6.0	93.5	7.0	95.2	4.0
ViT-NeT	✓	✓	✗	91.6	-	93.6	-	-	-
ST-ProtoPNet	✓	✓	✗	86.1	-	92.7	-	-	-
KMEx	✓	✓	✓	85.0	1.0	59.4	1.0	94.0	1.0
KCCs (ours)	✓	✓	✓	82.2	2.2	44.8	2.3	90.0	2.3

Table 2: Evaluation of accuracy and complexity of numerous SEMs across three widely used benchmark datasets. Due to architectural constraints and training vs. no-training, a direct comparison is challenging, but all ViT-based model in this table are based on a ViT with DinoV2 weights.

Quantitative and qualitative evaluation Table 2 shows the quantitative evaluation of KCC and numerous SEM baselines. In this case, a ViT-based model with pretrained weights from DinoV2 (Oquab et al., 2024) are used. Due to architectural constraints and training vs. no-training, a direct comparison is challenging. However, the results show that training-free methods can achieve comparable or better performance to supervised SEMs for some datasets. For example, KCC has better accuracy than ProtoPNet on CUB200, despite not being trained on this dataset at all. The most direct comparison for KCCs is KMEx, which achieved superior performance but at the cost of worse explainability, as shown in Table 1. Also note that for the CARS dataset, the training-free methods struggle, indicating that for fine-grained classification with highly specific features, supervision is still necessary.

Towards reduction of reader bias with automatic text description of keypoints A universal challenge for all XAI methods and their visualization is that the reader must interpret the explanations. This introduces a reader bias (Bove et al., 2024), since users can interpret the visualizations differently. Here, we show how KCCs can reduce this reader bias by taking advantage of recent ViTs with vision-language capabilities (Sun et al., 2023). The recent VLParts model (Sun et al., 2023) offers a general purpose foundation model that segments images into parts, where each part is automatically labeled with a text description from an open vocabulary. This removes the ambiguity in the interpretation of the visualizations and explicitly states what is being matched. We replace the part segmentation model of a KCC with the VLParts and follow the general methodology of KCCs.

Figure 1 shows how the introduction of ViT foundation models with vision-language capabilities can add an additional benefit to KCCs, namely automatic labeling of the keypoints. This removes the reader bias since the meaning of the keypoints is now explicit. We also evaluate the performance of this approach on the CUB200 dataset which achieves an accuracy of 82.7 and a complexity of 3, thus comparable performance to models in Table 2. We believe that the combination of KCCs and vision-language modeling offers a promising direction forwards for reducing reader bias in XAI.

6 DISCUSSION

Equal weighting of keypoints A key element of KCCs is that each keypoint is weighted equally, with the motivation that the reader only needs to be able to count to understand the explanation. However, it is evident that there will be keypoints with higher importance to a particular class, for example a highly class-specific beak shape in the bird classification setting. A straight-forward way to incorporate such information could be to weight the keypoints e.g. by the distance between keypoints. However, we deliberately avoid this as it will make understanding the decision significantly less intuitive, since the user must asses an arbitrary weighting factor without an inherent meaning. An alternative approach would be to identify class-specific keypoints and remove keypoints that

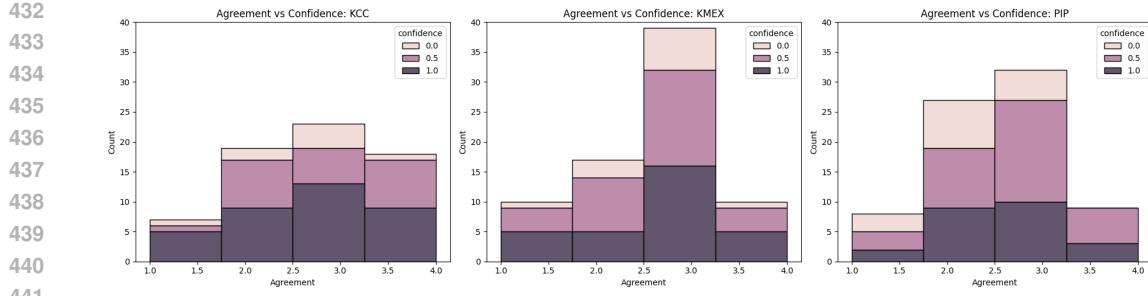


Figure 4: Results from user study on user confidence as a function of agreement with explanation. The results show that KCCs allows people to be more confident in correcting the model.

are shared among many classes to remove redundant keypoints. However, such a procedure is not straight-forward and we envision this as an interesting line of future research.

Findings from user study The results of the user study in Table 1 shows many interesting findings. User preferences is important since it indicates that users feel comfortable working with the explanations. But the user preference must always be seen in light of the quantitative measurements, since an algorithm can have a high user friendliness but still lead the user to make wrong decision. In this regards, KCC is superior to KMEx, where a high user preference is combined with improved quality and understanding of explanations.

Another important aspect is the issue of automation bias (Skitka et al., 1999), in which human have a tendency to trust the predictions of automatic systems. This automation bias is visible for all methods in Table 1, where participant struggle to identify incorrectly classified samples, which is in line with prior studies Kim et al. (2022). However, the participants were also asked; "do you feel confident to make the right prediction based on the explanation shown?", which is not shown in Table 1. This is important, because a step towards reducing automation bias is enabling users to confidently agree/disagree with the model. Figure 4 shows the confidence score of the user as a function of agreement with the model. When users disagree with a prediction, they select an agreement of 1 or 2, and in those cases, KCC has higher confidence scores compared to KMEx and PIP. This indicates that KCCs allows people to be more confident in correcting the model.

7 CONCLUSION

We present a new paradigm for self-explainable deep learning that requires no training, is suitable for ViTs, and provide a new way of visualizing explanations. A user study showed that KCCs allowed humans to understand the explanations to a greater extent compared to existing approaches. We also showed how exploiting recent advances in vision-language modeling could provide automatic text description towards reducing reader bias. We believe that KCCs provides a completely new direction within XAI with great potential to improve the transparency of deep learning.

REFERENCES

- Radhakrishna Achanta, Appu Shaji, Kevin Smith, et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 2012. doi: 10.1109/TPAMI.2012.120.
- Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCVW What is Motion For?*, 2022.
- Ananthu Aniraj, Cassio F Dantas, Dino Ienco, et al. Pdiscoformer: Relaxing part discovery constraints with vision transformers. In *ECCV*, 2024. doi: https://doi.org/10.1007/978-3-031-73013-9_15.
- Shahaf Bassan, Ron Eliav, and Shlomit Gur. Explain yourself, briefly! self-explaining neural networks with concise sufficient reasons. In *ICLR*, 2025. URL <https://openreview.net/forum?id=8nuzsfiQfS>.

486 Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94, 1987. doi: 10.1037/0033-295x.94.2.115.

487

488

489 Clara Bove, Thibault Laugel, Marie-Jeanne Lesot, Charles Tijus, and Marcin Detyniecki. Why do
490 explanations fail? a typology and discussion on failures in xai. *Arxiv (cornell University)*, 2024.
491 doi: 10.48550/arXiv.2405.13474.

492

493 Zachariah Carmichael, Suhas Lohit, Anoop Cherian, et al. Pixel-Grounded Prototypical Part Net-
494 works . In *WACV*, 2024. doi: 10.1109/WACV57701.2024.00470.

495

496 Chaofan Chen, Oscar Li, Chaofan Tao, et al. This looks like that: deep learning for inter-
497 pretable image recognition. In *NeurIPS*, 2019. URL <https://dl.acm.org/doi/10.5555/3454287.3455088>.

498

499 Omid Davoodi, Shayan Mohammadizadehsamakosh, and Majid Komeili. On the interpretability
500 of part-prototype based classifiers: a human centric analysis. *Scientific Reports*, 2023. doi:
10.1038/s41598-023-49854-z.

501

502 Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable ProtoPNet: An Interpretable
503 Image Classifier Using Deformable Prototypes . In *CVPR*, 2022. doi: 10.1109/CVPR52688.
2022.01002.

504

505 Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. An image is worth 16x16 words:
506 Transformers for image recognition at scale. In *ICLR*, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.

507

508 Srishti Gautam, Ahcene Boubekki, Stine Hansen, et al. Protovae: a trustworthy self-explainable
509 prototypical variational model. In *NeurIPS*, 2022. URL <https://dl.acm.org/doi/10.5555/3600270.3601574>.

510

511 Srishti Gautam, Ahcene Boubekki, Marina MC Höhne, et al. Prototypical self-explainable mod-
512 els without re-training. *TMLR*, 2024. URL <https://openreview.net/forum?id=HU5DOUp6Sa>.

513

514

515 Kai Han, Yunhe Wang, Hanting Chen, et al. A survey on vision transformer. *IEEE TPAMI*, 2023.
516 doi: 10.1109/TPAMI.2022.3152247.

517

518 Adrian Hoffmann, Claudio Fanconi, Rahul Rade, et al. This looks like that... does it? shortcomings
519 of latent space prototype interpretability in deep networks. In *ICML Workshop on Theoretic
520 Foundation, Criticism, and Application Trend of Explainable AI*, 2021.

521

522 Sangwon Kim, Jaeyeal Nam, and Byoung Chul Ko. ViT-NeT: Interpretable vision transformers
523 with neural tree decoder. In *ICML*. URL <https://proceedings.mlr.press/v162/kim22g.html>.

524

525 Sunnie S. Y. Kim, Nicole Meister, Vikram V. Ramaswamy, et al. HIVE: Evaluating the human
526 interpretability of visual explanations. In *ECCV*, 2022.

527

528 Sunnie S. Y. Kim, Elizabeth Anne Watkins, Olga Russakovsky, et al. "help me help the ai":
529 Understanding how explainability can support human-ai interaction. In *CHI*, 2023. doi:
10.1145/3544548.3581001.

530

531 Alexander Kirillov, Eric Mintun, Nikhila Ravi, et al. Segment anything. *arXiv:2304.02643*, 2023.

532

533 Jonathan Krause, Michael Stark, Jia Deng, et al. 3d object representations for fine-grained catego-
534 rization. In *ICCV Workshops*, 2013. doi: 10.1109/ICCVW.2013.77.

535

536 Satyapriya Krishna, Tessa Han, Alex Gu, et al. The disagreement problem in explainable machine
537 learning: A practitioner's perspective. *TMLR*, 2024. URL <https://openreview.net/forum?id=jESY2WTZCe>.

538

539 Luca Longo, Mario Brcic, Federico Cabitza, et al. Explainable artificial intelligence (xai) 2.0: A
manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 2024.
doi: <https://doi.org/10.1016/j.inffus.2024.102301>.

540 Richard E. Mayer. Designing multimedia instruction in anatomy: An evidence-based approach.
 541 *Clinical Anatomy*, 2018. doi: 10.1002/ca.23265.

542

543 Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-
 544 grained image recognition. In *CVPR*, 2021.

545 Meike Nauta, Jörg Schlötterer, Maurice van Keulen, et al. Pip-net: Patch-based intuitive prototypes
 546 for interpretable image classification. In *CVPR*, June 2023.

547

548 Frank H Netter. *Atlas of Human Anatomy*. Netter Basic Science. Elsevier, 7 edition, December
 549 2017.

550 Maxime Oquab, Timothée Darcet, Théo Moutakanni, et al. DINOv2: Learning robust visual fea-
 551 tures without supervision. *TMLR*, 2024. URL <https://openreview.net/forum?id=a68SUT6zFt>.

552

553 Shaul Oron, Tali Dekel, Tianfan Xue, et al. Best-Buddies Similarity—Robust Template Matching
 554 Using Mutual Nearest Neighbors. *TPAMI*, 2018. doi: 10.1109/TPAMI.2017.2737424.

555

556 Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Cats and dogs. In *CVPR*, 2012.

557

558 Olin Sewall Pettingill. Topography. In Olin Sewall Pettingill (ed.), *Ornithology in Laboratory and*
 559 *Field (Fourth Edition)*, pp. 10–28. Academic Press, fourth edition edition, 1984. ISBN 978-0-12-
 560 552450-6. doi: <https://doi.org/10.1016/B978-0-12-552450-6.50006-0>. URL <https://www.sciencedirect.com/science/article/pii/B9780125524506500060>.

561

562 Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang,
 563 Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing
 564 Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks,
 565 2024.

566

567 Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and
 568 Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In
 569 *European Conference on Computer Vision*, pp. 351–368. Springer, 2022.

570 Linda J. Skitka, Kathleen L. Mosier, and Mark Burdick. Does automation bias decision-making? *In-*
 571 *ternational Journal of Human-Computer Studies*, 51(5):991–1006, November 1999. ISSN 1071-
 572 5819. doi: 10.1006/ijhc.1999.0252. URL <http://dx.doi.org/10.1006/ijhc.1999.0252>.

573

574 Peize Sun, Shoufa Chen, Chenchen Zhu, Fanyi Xiao, Ping Luo, Saining Xie, and Zhicheng Yan. Go-
 575 ing denser with open-vocabulary part segmentation. In *2023 IEEE/CVF International Conference*
 576 *on Computer Vision (ICCV)*, pp. 15407–15419, 2023. doi: 10.1109/ICCV51070.2023.01417.

577

578 Hugues Turbé, Mina Bjelogrlic, Gianmarco Mengaldo, and Christian Lovis. Protos-vit: Visual
 579 foundation models for sparse self-explainable classifications, 2024.

580 C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds-200-2011. Techni-
 581 cal Report CNS-TR-2011-001, California Institute of Technology, 2011.

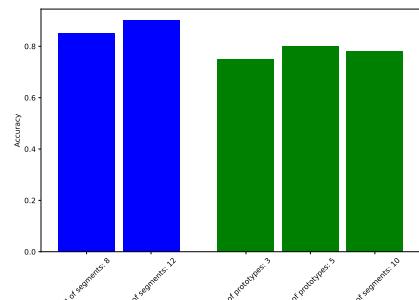
582

583 Chong Wang, Yuyuan Liu, Yuanhong Chen, Fengbei Liu, Yu Tian, Davis McCarthy, Helen Frazer,
 584 and Gustavo Carneiro. Learning support and trivial prototypes for interpretable image classifica-
 585 tion. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2062–2072,
 586 2023. doi: 10.1109/ICCV51070.2023.00197.

587 Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by con-
 588 structing transparent embedding space. In *2021 IEEE/CVF International Conference on Com-*
 589 *puter Vision (ICCV)*, pp. 875–884, 2021. doi: 10.1109/ICCV48922.2021.00093.

590

591 Romain Xu-Darme, Georges Quénot, Zakaria Chihani, and Marie-Christine Rousset. Sanity checks
 592 for patch visualisation in prototype-based image classification. In *2023 IEEE/CVF Conference*
 593 *on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3691–3696, 2023. doi:
 10.1109/CVPRW59228.2023.00377.

594 A QUALITATIVE EXAMPLES
595596 Two supplementary examples of KCCs with automatic text descriptions are presented to illustrate
597 the generalizability of our results.
598613 Figure 5: Automated keypoints description using vision-language model
614
615616 B HYPERPARAMETERS
617618 This section shows results for different number of segments, different number of prototypes, and
619 with different encoder
620631 Figure 6: Different number of segments
632 and prototypes.
633

method	encoder	CUB200	CARS	PETS
KMEx	clip	61.0	68.3	75.0
KCCs (ours)	clip	57.6	64.0	75.4

634 Table 3: KCC with CLIP encoder.
635
636
637
638
639
640
641
642
643
644
645
646
647