Reviewer f9CQ

1. Analysis with ResNet Architecture

The paper considers mostly convolutional networks. It lacks analysis of more state-of-the art architectures like residual networks or transformers.

As suggested by the reviewer, we extended our analysis to a ResNet-50 architecture. We fine-tuned the ResNet model on the CUB dataset and reproduced the experiment fro Section 3.1.1 of our paper with this new architecture. In particular, we fit a CAR and a CAV classifier on the penultimate layer of the ResNet. We report the accuracy averaged over the $C = 112$ CUB concepts bellow.

ResNet Layer	CAR Accuracy (mean $\pm$ sem)	CAV Accuracy (mean $\pm$ sem)
Layer4	.89 $\pm$ .01	.87 $\pm$ .01

As we can see, CAR classifiers are highly accurate to identify concepts in the pnultimate ResNet layers. As in our paper, we observe that CAR classifiers outperform CAV classifiers, although the gap is smaller than for the Inception-V3 neural network. We deduce that our CAR formalism extends beyond the architectures explored in the paper.

Copy the NLP experiment from reviewer qiiw.

2. Increasing Explainability at Training Time

Would it be possible to get inspiration from the insights in the paper to improve neural network training, with the purpose of increasing explainability?

Propose a regularization/pretraining scheme? If time permits, do a small experiment.

3. Sensitivity to Adversarial Attacks

How are the proposed explanations sensitive to adversarial attacks?

As suggested by the reviewer, we perform an experiment to evaluate the robustness of CAR explanations with respect to adversarial perturbations. In this experiment, we work with the MNIST dataset in the same setting as the experiment from Section 3.1.2 from our paper. We train a CAR concept classifier for each MNIST concept $c \in [C]$. We use the CAR classifier to output TCAR scores relating the concept c with each class $k \in [d_Y]$. As in the main paper, since the ground-truth association between concepts and classes is known (e.g. the class corresponding to digit 8 will always have the concept loop), we can compute the correlation $r(\mathrm{TCAR}, \mathrm{TrueProp})$ between our TCAR score and the ground-truth proportion of examples that exhibit the concept. In this experiment, this correlation is evaluated on a test set $\mathcal{D}_{\mathrm{test}} = \mathcal{D}_{\mathrm{adv}} \sqcup \mathcal{D}_{\mathrm{orig}}$ that contains adversarial test examples $\mathcal{D}_{\mathrm{adv}}$ and original test examples $\mathcal{D}_{\mathrm{orig}}$. Each adversarial MNIST image $x_{\mathrm{adv}} \in \mathcal{D}_{\mathrm{adv}}$ is constructed by finding a small (w.r.t. the $\| \cdot \|_\infty$ norm) perturbation $\epsilon \in \mathbb{R}^{d_X}$ around an original test image $x \in \mathcal{X}$ that maximizes the prediction shift for the black-box $f : \mathcal{X} \to \mathcal{Y}$:

$$\epsilon = \arg \max_{\tilde{\epsilon} \in \mathbb{R}^{d_X}} \mathrm{CrossEntropy}[f(x), f(x + \tilde{\epsilon})] \ s.t. \ \|\tilde{\epsilon}\|_\infty < .1$$

The adversarial image is then defined as $x_{\mathrm{adv}} \equiv x + \epsilon$. We measure the correlation $r(\mathrm{TCAR}, \mathrm{TrueProp})$ by varying the proportion $\frac{|\mathcal{D}_{\mathrm{adv}}|}{|\mathcal{D}_{\mathrm{test}}|}$ of adversarial examples in the test set. The results are reported bellow.

Adversarial %	$r(\mathrm{TCAR}, \mathrm{TrueProp})$
0	.99
5	.99
10	.99
20	.99
50	.97
70	.96
100	.92

We observe that the TCAR scores keep a high correlation with the true proportion of examples that exhibit the concept even when all the test examples are adversarially perturbed. We conclude that TCAR explanations are robust to adversarial perturbations in this setting.

For completeness, we have also adapted the background shift robustness experiment in Section 7 from *Koh, P. et al. (2020). Concept Bottleneck Models.* As in our paper, we use CAR to explain the predictions of our Inception-V3 model trained on the original CUB training set. The explanations are made on test images where the background has been replaced. As Koh et al., we use the segmentation of the CUB dataset to isolate the bird on each image. The rest of the image is replaced by a random background sampled from the *Place365* dataset. This results in a test set $\mathcal{D}_{\mathrm{test}}$ with a background shift with respect to the training set. By following the approach from Section 3.1.2 of our paper, we measure the correlation $r(\mathrm{TCAR}, \mathrm{TrueProp})$ between the TCAR score and the true proportion of examples in the class that exhibit the concept for each $(\mathrm{class}, \mathrm{concept})$ pair. We measured a correlation of $r(\mathrm{TCAR}, \mathrm{TrueProp}) = .82$ in the background-shifted test set. This is close to the correlation for the original test set reported in the main paper, which suggests that CAR explanations are robust with respect to background shifts. Note that this correlation is still better than the one obtained with TCAV on the original test set.

Reviewer znZB

1. Hyperparameter Choice

The introduction of kernels may lead to much more hyper-parameter choice to use in practice for very complicated concepts and networks, an additional RBF kernel may be insufficient to make the data linearly separable in practice.

Since our CAR classifiers are kernel-based, they indeed come with extra hyperparameters (e.g. the kernel width). We would like to emphasize that, to ensure fair comparisons with the CAV classifiers, none of these hyperparameter has been optimized in the experiments from Section 3.1.1 and 3.1.2. We have used the default hyperparameters in the scikit-learn implementation of support vector classifiers. In all our experiments, CAR classifiers substantially outperform CAV hyperparameters without having to tune the hyperparameters.

In the case where the user desires a CAR classifier that generalizes as well as possible, tuning these hyperparameters might be useful. We propose to tune the hyperparameters θ_h of our CAR classifiers s^c_κ for each concept $c \in [C]$ by using Bayesian optimization and a validation concept set:

1. Randomly sample the hyperparameters from an initial prior distribution $\theta_h \sim P_{\text{prior}}$.
2. Split the concept sets $\mathcal{P}^c, \mathcal{N}^c$ into training concept sets $\mathcal{P}^c_{\text{train}}, \mathcal{N}^c_{\text{train}}$ and validation concept sets $\mathcal{P}^c_{\text{val}}, \mathcal{N}^c_{\text{val}}$.
3. For the current value θ_h of the hyperparameters, fit a model s^c_κ to discriminate the training concept sets $\mathcal{P}^c_{\text{train}}, \mathcal{N}^c_{\text{train}}$.
4. Measure the accuracy $\text{ACC}_{\text{val}} = \frac{\sum_{x \in \mathcal{P}^c_{\text{val}}} \mathbf{1}(s^c_\kappa \circ g(x)=1) + \sum_{x \in \mathcal{N}^c_{\text{val}}} \mathbf{1}(s^c_\kappa \circ g(x)=0)}{|\mathcal{P}^c_{\text{val}} \cup \mathcal{N}^c_{\text{val}}|}$
5. Update the current hyperparameters θ_h based on ACC_{val} using Bayesian optimization (Optuna in our case).
6. Repeat 3-5 for a predetermined number of trials.

We applied this process to the CAR accuracy experiment (same setup as in Section 3.1.1 of the main paper) to tune the CAR classifiers for the CUB concepts. Interestingly, we noticed no improvement with respect to the CAR classifiers reported in the main paper: tuned and standard CAR classifier have an average accuracy of $(93 \pm .2)\%$ for the penultimate inception layer. This suggests that the accuracy of CAR classifiers is not heavily dependant on hyperparameters in this case. That said, we believe that the above approach to tune the hyperparameters of CAR classifiers might be useful in other cases so we will add it to the manuscript.

We agree that not all concepts can be captured by CAR classifiers, even after hyperparameter optimization (e.g. Figure 4.c from the paper, we see that CAR classifiers don't generalize well on the layer Mixed-5d). As we argue in the paper, this is a strong indication that our concept smoothness assumption (Assumption 2.1) is violated. This implies that concepts are not smoothly encoded in the geometry of the model's representation space $\mathcal{H}$. The user can therefore deduce that the concept is unlikely to be salient to interpret $\mathcal{H}$. In that sense, the inability to fit CAR classifiers that generalize well is as informative as the ability to fit CAR classifiers that generalize well. Note that this whole reasoning is made more quantitative through statistical hypothesis testing in Section 3.1.1 of our paper.

2. Layer Selection

The layer selection problem is not dealt with, as TCAV suggested that sometimes earlier layers may be more fruitful even if the accuracy seems lower (for more low-level concepts). In my personal experience, using the mixed-7b layer inception-V3 for TCAV usually produces a much better result than the penultimate layer. Moreover, how does one choose the layer to apply TCAR is not addressed. note:TCAV does not work well with the penultimate layer since d c/d activation = W_c (which is independent to the instance), and thus for all instances in the same class the directional derivative to the same concept would be fixed.

We would like to emphasize our CAR formalism, the user is free to to choose the layer they want to interpret. In many use cases, the user might want to interpret specific layers of the neural network based on their knowledge of the architecture. In creating TCAR explanations for various layers, we decided to select the layer for which the concept classifiers (for both CAR and CAV) generalize better to unseen examples, as measured in our experiment from Section 3.1.1 from our paper.

Following the reviewer's recommendation, we decided to repeat the comparison beween TCAV and TCAR from Section 3.1.2 with the layer Mixed-7b of our Inception-V3 classifier. In doing so, we measured the following correlation between the scores and the groundtruth proportion of examples within a class that exhibit the concept:

$$r(\mathrm{TCAV}, \mathrm{TrueProp}) = .46 \qquad\qquad r(\mathrm{TCAR}, \mathrm{TrueProp}) = .71$$

For both TCAV and TCAR, these correlations are lower than the ones obtained in the model's penultimate layer. In this case, it appears that the association between classes and concepts are more meaningfully encoded in the deeper layers of the neural network. We believe that the machine learning community would greatly benefit from the ability to perform this type of analysis for various architectures.

3. Discrepancy between CAV and TCAV Accuracy

In the inception experiment, CAV has a pretty close accuracy to CAR (showing that the linear separability is not a huge issue), but the

evaluation of TCAV score in table 1 seems completely wrong. The failure of TCAV in MNIST and ECG is understandable since the model seems under-represented, and an additional kernel classifier would help a lot. However, the result in Inception-v3 is not convincing.

We would like to thank the reviewer for pointing this out. After double checking our implementation, everything seems consistent and we are confident about the results reported in our paper. We would like to emphasize that the accuracy of a CAV concept classifier does not guarantee that TCAV score correlates well with the ground-truth association between classes and concepts. This can be understood in the following way:

- A highly accurate CAV classifier occurs when the concept sets are linearly separable in the model's representation space $\mathcal{H}$. This means that the feature extractor $g : \mathcal{X} \to \mathcal{H}$ tends to linearly separate examples that exhibit the concept from the ones that don't.
- A high correlation between the TCAV score and the ground-truth association between classes and concept occurs when the model prediction for each class is sensitive to the presence of the appropriate concepts. This means that for each class $k \in [d_Y]$, the label map $l_k : \mathcal{H} \to [0, 1]$ is sensitive to concepts $c \in [C]$ that are truly relevant to identify this class (e.g. MNIST images of digit 9 are sensitive to the loop concept).

From the above discussion, we immediately notice that the two previous situations depend on two orthogonal parts of the model: the accuracy of the CAV classifier depends on the feature extractor g and the correlation of the TCAV score with ground-truth depends on the label map l. In that light, these two situations appear independent from each other: it is perfectly possible to have highly accurate CAV classifiers and poor TCAV scores if concepts are well separated in the model's representation space $\mathcal{H}$ but the model's predictions are not sensitive to the right concepts. We note that this is precisely what occurs in the CUB setting, as we can observe from Figures 5.c in the main paper and Figure 15 in the supplementary material. In these figures, we observe that TCAV suggests non-existent associations, such as one between the class *black crow* and the concept *yellow wing colour*.

A possible explanation for the better agreement between the quality of CAR classifiers and TCAR scores is the fact that, unlike TCAV, TCAR scores are not computed by using the sensitivity metric S_k^c defined in Section 2.1 of the paper. As explained in Section 2.2, we use the concept activation regions *directly* to compute TCAR scores. This implies that TCAR scores are computed in the model's representation space $\mathcal{H}$ *directly* by analyzing how different classes are scattered across the concept clusters. We believe that this different characterization might explain the gap between TCAV and TCAR scores in terms of correlation with the ground-truth. This would suggest that TCAV's sensitivity S_k^c might not be the most appropriate way to detect the association between a class and a concept. We will make sure to add this discussion in the manuscript.

4. Significance of Feature Importance Evaluation

The evaluation of concept-based feature importance is only a sanity check, how is this useful?

We believe that our consistency checks for concept-based feature importance demonstrate two crucial and non-trivial points on various datasets:

1. **Concept-based saliency maps are not generic.** The low correlation between vanilla saliency maps and concept-based saliency maps indicates that the latter are concept-specific. This is consistent with the fact that the features that are salient to identify a concept are not necessarily the same as the ones that are salient to make a prediction.
2. **Concept-based saliency maps are consistent with human intuition.** The correlation between the saliency maps of each pair of concept (c_1, c_2) appears to be important when c_1 and c_2 can be identified through the same input features (e.g. the *loop* and the *curvature* concepts in MNIST are both identified through pixels in the curved part of the digit). This is a way to confront our concept-based saliency maps with the ground-truth human knowledge, in the same spirit as the assessment of global explanations in Section 3.1.2. We note that the former are more difficult in practice since no concept-specific ground-truth saliency map is available for the investigated dataset. This is why we use correlation to verify that saliency maps correlations are consistent with the human characterization of those concepts.

Reviewer qiiw

1. Generalizing CAV Sensitivity Interpretations

One of the critical weaknesses of CAR is that since explanations are generated through a Kernel-based technique, it loses the nice interpretations CAV offers, like if one increases the presence of a concept, how does it affect the model predictions. Would it be possible to replicate the interpretations CAV offers like concept sensitivity? An alternate would be to perform test time interventions on the Kernel-based Concept classifier.

We thank the reviewer for suggesting this interesting extension. In our formalism, it is perfectly possible to define a *local* concept activation vector through the concept density $\rho^c : \mathcal{H} \to \mathbb{R}^+$ defined in Definition 2.1 from the main paper. Indeed, the vector $\nabla_h \rho^c[h] \in \mathcal{H}$ points in the direction of the representation space $\mathcal{H}$ where the concept density (and hence the presence of the concept) increases. Hence, this vector can be interpreted as a *local* concept activate vector. Note that this vector becomes global whenever we parametrize the concept density ρ^c with a linear kernel $\kappa(h_1, h_2) = h_1^\intercal h_2$. Equipped with this generalized notion of concept activation vector, we can also generalize the CAV concept sensitivity

S_k^c by replacing the CAV w^c by $\nabla_h \rho^c[h]$ for the representation $h = g(x)$ of the input $x \in \mathcal{X}$:

$$S_k^c(x) \equiv (\nabla_h \rho^c[g(x)])^\intercal (\nabla_h l_k[g(x)]).$$

In this way, all the interpretation provided by the CAV formalism are also available in the CAR formalism. We will make sure to add this discussion in the manuscript.

2. Using CAR with Unsupervised Concepts

Is it possible to run CAR with concepts (or latent variables) generated by Autoencoders or VAE that are inherently noisy or abstract? Could CAR be used to analyze the concepts generated by techniques like SENN?

Our CAR formalism adapts to a wide variety of neural network architecture. As suggested by the reviewer, we use CAR to analyze the concepts discovered by a self explaining neural network (SENN) trained on the MNIST dataset. As in *Alvarez-Melis, D., & Jaakkola, T. (2018). Towards Robust Interpretability with Self-Explaining Neural Networks.*, we use a SENN of the form

$$f(x) = \sum_{s=1}^{S} \theta_s(x) \cdot g_s(x),$$

Where $h_s(x)$ and $\theta_s(x)$ are respectively the activation and the relevance of the synthetic concept $s \in [S]$ discovered by the SENN model. We follow the same training process as Alvarez-Melis et al. This yields a set of $S = 5$ concepts explaining the predictions made by the SENN $f : \mathcal{X} \to \mathcal{Y}$.

We use our CAR formalism to study how the synthetic concepts $s \in [S]$ discovered by the SENN are related to the concepts $c \in \{\text{Loop}, \text{Vertical Line}, \text{Horizontal Line}, \text{Curvature}\}$ introduced in our paper. In our formalism, the relevance of a concept c for a given prediction $x \mapsto f(x)$ is measured by the concept density $\rho^c \circ g(x)$. To analyze the relationship between the SENN concept s and the concept c, we can therefore compute the correlation of their relevance:

$$r(s, c) = \text{corr}_{X \sim P_{\text{empirical}}(\mathcal{D}_{\text{test}})}[\theta_s(X), \rho^c \circ (X)].$$

The larger this correlation, the more concept s and c tend to be relevant together. We report the correlation between each pair (s, c) in the bellow table.

Correlation $r(s, c)$	Loop	Vertical Line	Horizontal Line	Curvature
SENN Concept 1	-0.28	-0.12	0.26	0.11
SENN Concept 2	-0.50	0.71	-0.03	-0.69
SENN Concept 3	-0.47	0.10	0.71	-0.14
SENN Concept 4	-0.33	0.02	-0.06	-0.01
SENN Concept 5	0.57	-0.0	-0.63	0.07

We note the following:

1. SENN Concept 2 correlates well with the Vertical Line Concept.
2. SENN Concept 3 correlates well with the Horizontal Line Concept
3. SENN Concept 5 correlates well with the Loop Concept.
4. SENN Concepts 1 and 4 are not well covered by our concepts.

The above analysis shows the potential of our CAR explanations to better understand the abstract concepts discovered by SENN models. We believe that the community would greatly benefit from the ability to perform similar analyses for other interpretable architectures, such as disentangled VAEs.

3. Robustness of CAR Explanations

How robust are explanations generated by CAR? Is it robust to change in backgrounds of the images?

Copy and adapt the robustness point from reviewer f9CQ.

4. CAR for NLP

Would CAR work for other domains like NLP?

CAR is a general framework and can be used in a wide variety of domains that involve neural networks. In our paper, we show that CAR provides explanations for various modalities: 1. Large image dataset 2. Medical time series 3. Medical tabular data.

As suggested by the reviewer, we perform a small experiment to assess if those conclusions extend to the NLP setting. We train a small CNN on the IMDB Review dataset to predict whether a review is positive or negative. We use Glove to turn the word tokens into embeddings. We would like to assess whether the concept $c =$ Positive Adjective is encoded in the model's representations. Examples that exhibit the concept c are sentences containing positive adjectives. We collect a positive set $\mathcal{P}^c$ of $N^c = 90$ such sentences. The negative set $\mathcal{N}^c$ is made of N^c sentences randomly sampled from the Gutenberg Poem Dataset. We verified that the sentences from $\mathcal{N}^c$ did not contain positive adjectives. We then fit a CAR classifier on the representations obtained in the penultimate layer of the CNN.

We assess the generalization performance of the CAR classifier on a holdout concept set made of $N^c = 30$ concept positive and negative sentences (60 sentences in total). The CAR classifier has an accuracy of 87% on this holdout dataset. This suggests that the concept c is smoothly encoded in the model's representation space, which is consistent with the importance of positive adjectives to identify positive reviews. We deduce that our CAR formalism can be used in a NLP setting. We believe that using CAR to analyze large-scale language model would be an interesting study that we leave for future work.

5. Using CAR without Human Annotations

Is it possible to relax the assumption that such a class of techniques requires additional annotation of concepts? As of now, CAR requires a user to specify the positive and negative examples for each concept.

Discuss the work by Ghorbani.

6. Minor Points

Given that CAR assumes to have access to the feature extractor of the model, it isn't truly a black-box setup, unlike paper portrays.

I would encourage the authors to list the limitations of the proposed approach.

We thank the reviewer for these additional remarks. We will make sure that to implement those changes in the final manuscript.

Experiments TODO

- DONE *Small* Kernel width optimization
- DONE *Medium* CUB with ResNet
- DONE *Small* CUB TCAR/TCAV and Acc with mixed-7b layer
- DONE *Large* NLP Experiment
- DONE *Small* Robustness to adversarial attacks
- DONE *Medium* CUB with background shift
- DONE CAR and unsupervised concepts?