We thank all reviewers for their constructive comments. We first address some general questions, and then respond to the questions raised by each reviewer separately.

**Layer Choice (R2,R3)** The layer can be chosen depending on the size of the nearest neighbor patch the user would like to visualize, since this size depends on the receptive field of the feature layer. Deeper layers have larger receptive fields while shallower layers have smaller receptive field size. In AwA experiments, we choose Mixed_5d layer so that the receptive field is $127 \times 127$, which is half of the original image size and capable of capturing both larger and smaller concepts. If the user is only interested in smaller and more low level concepts (such as the texture of image), we can apply our method to earlier layers. As a hyper-parameter control experiment, we apply our method to mixed_5c layer where the receptive field is $95 \times 95$ and visualize the result below in Fig. 1 top right, where indeed smaller (low level) concepts such as eyes (in constrast to head), furry, sandy are captured (which we name the concepts). If one finds the concepts discovered to be too small (low level), they could apply the method on a deeper layer and vice versa.

**Qualitative Results for Baselines and User study on AwA (R1,R3):** We have added qualitative examples (see Fig. 1 bottom row) for concepts discovered by PCA and Kmeans on AwA (ACE gives the same set of concepts as Kmeans if superpixels are replaced by squares), where we choose top concepts by conceptSHAP (in the same way as our method exactly) for the classes "squirrel, rabbit, bobcat". To evaluate our method, PCA, Kmeans (ACE is Kmeans without superpixel), we added a human study suggested by R3 on AwA. For each method, we randomly choose 1 top concept per class for the 3 classes, and thus we choose 3 concepts per method (9 concepts in total). For each concept, we show users 4 top images of that concept, and ask users to choose the image (out of 3 different options) that they believe should belong to the same concept (where one of the option will actually belong to the same concept, and the other two are random image patches of the same class that does not belong to that concept). We then calculate the average accuracy to measure the human interpretability of the concept discover method. We conduct a user study with 10 users, where each of them are asked with the same 9 questions (1 question per concept chosen). The average correct ratio for our method, PCA, and Kmeans are 0.733, 0.267, and 0.6 respectively, showing our method's superiority. Kmeans outperforms PCA as it also encourages closeness of top nearest neighbors (which is better for ostensive definition).

**Hyperparameters (R2):** To choose the hyperparameters, one can use a small-scale evaluation dataset to choose a few important hyperparameters. One should choose the hyperparamters so that they get concepts with high completeness and $R_1(\mathbf{c})$, and we better describe the impact of these hyperparamters to guide such selection. We provide experiments on sensitivity to hyperparameters in the appendix: to the number of concepts in Fig. 2, and to $\lambda, \beta$ in Figs. 4, 5 and 6 in Appendix. We fixed the architecture $g$ in our previous experiments to a two layer neural network. It can also be a linear network followed by the remaining neural network $h$ (so that $h$ is also optimized in eq.3 in line 209). We show the qualitative results of AwA where $g$ includes $h$ (which we call "ours-linear g+h") in Fig. 1 (top left), where we obtain interpretable concepts (we named the concepts) with high completeness (thus the hyperparameter is not sensitive to architecture if $g$ has enough representative power). We discussed how to select the layer choice and its impact above.

**Optimization, Computation (R2):** To optimize the objective $\arg\max_{\mathbf{c}_{1:m},g} \log P(h_y(g(v_{\mathbf{c}}(x_{1:T}))) + R(\mathbf{c})$, we optimize variables $\mathbf{c}_{1:m}, g$ jointly. When the optimization converges, $g$ is a (local) optimal value given $\mathbf{c}_{1:m}$. The computational cost for discovering concepts and calculating conceptSHAP is about 3 hours for AwA dataset and less than 20 minutes for the toy dataset and IMDB, using a single 1080 Ti GPU, which can be further accelerated with parallelism. The computational cost is low since the pretrained model is fixed, and we only optimize for $g$ and $\mathbf{c}$.

**R3:** The point in lines 117-119 pointed out by the reviewer is indeed a typo, it should be $x$ instead of $x_t$, we have corrected it. While XOR may make the ground truth concept difficult to discover, in our toy example we show that our method can still retrieve the correct concepts with XOR in the model. Since we show that our method can work on both toy and real datasets, we believe this further demonstrates the generality of our method even when the optimization may seem difficult. The top-K loss does encourages human interpretability by allowing ostensive definition, which we will better discuss. For the NLP experiment, we added a control scenario where we append 5 random subsentences, and the average prediction score becomes 0.498 (oiginally 0.516). Thus, appending discovered concepts experiments is not caused by being out-of-distribution. We didn't test on Imagenet since we can't visualize results for all 1000 classes.

Figure 1: Visualizing ours-linear g+h (top left), ours-mixed_5c (top right), PCA (bottom left), Kmeans (bottom right).