A Dataset and Framework for Learning State-invariant Object Representations

Supplementary Material

7. Additional details about the ObjectsWithStateChange (OWSC) dataset

In this section, we provide additional details about the dataset described in Sec. 3 of the main paper.

1. **State changes of the objects during data collection:** A comprehensive list of all possible state changes and other transformations for objects from each category is provided in Tbl. 6. Depending on the physical characteristics of the specific object, all the state changes and transformations may not be applicable for each object, and images are collected if such state changes are possible for the objects of interest.
2. **Examples of Train and Test images from OWSC-SI split:** Some examples from both the train and test data from OWSC-SI split that comprise images of the object in different states captured from arbitrary viewpoints are shown in Fig. 9.
3. **Examples of Gallery and Probe images from OWSC-GN split**: Some examples from the OWSC-GN split are shown in Fig. 10. The gallery images comprise reference images of an object from arbitrary viewpoints. The probe images comprise of each object with transformations (placed in different lighting conditions, backgrounds, poses), and state changes captured from arbitrary points.
4. **Annotations:** Each object is annotated with a category label, object label, and a text description describing the visual characteristics (such as color, material, texture, shape, and other unique visual attributes) of the object to facilitate multi-modal representation learning, as shown in Fig. 11.

7.1. Discussion about the unique contributions of the dataset

Broader Impact and Future Research:

1. We believe that to the best of our knowledge, we are the first to introduce a dataset that incorporates complex state changes of objects that can be used for category-level and fine-grained object-level recognition and retrieval tasks. This could benefit several real-world applications, such as automatic checkout systems [13, 20], surveillance and automation systems [12], and robotic systems that deal with recognizing objects under various state changes upon interaction with them.
2. This dataset serves as a cross-dataset evaluation benchmark, enabling assessment of methods trained on other datasets for robustness to state changes and complex transformations and tested on our OWSC-GN split, as demonstrated in Sec. 8.
3. We provide text annotations describing the visual characteristics of objects that can be used for training multi-modal models and VLMs. This can be used for adapting such models on object image collections for learning transformation-invariant multi-modal representations and generalized zero-shot evaluation.
4. This dataset can be extended to support visual question answering (VQA) using vision-language models (VLMs) and large language models (LLMs), enabling

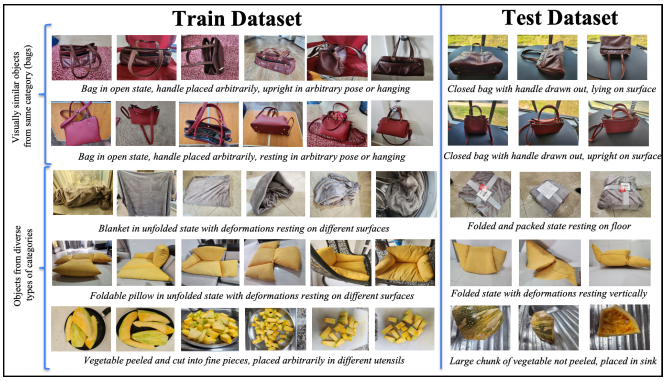

Figure 9. shows samples of different objects from the Train and Test data for the OWSC-SI split. The state of the object as well as the background and pose are different in each split for every object. The images are captured from arbitrary viewpoints. The first two rows show similar looking objects from the same category for fine-grained recognition and retrieval and the remaining rows show other objects from diverse categories.

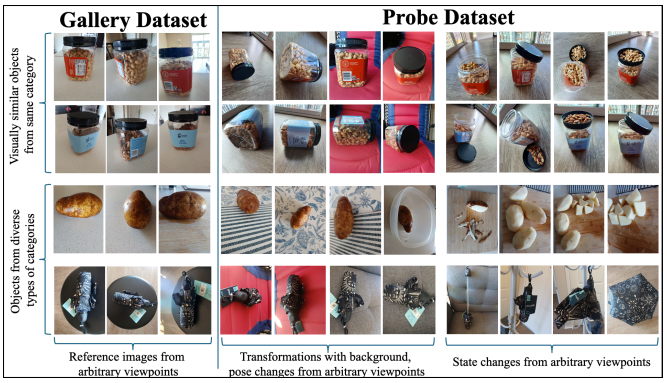

Figure 10. shows example images from the OWSC-GN split that comprises of novel (unseen) objects that are not present in the OWSC-SI split. The first two rows show similar looking objects from the same category for fine-grained recognition and retrieval and the last two rows show other objects from diverse categories.

Category	Category-specific state changes
bags	open and close bag, empty bag and fill it with items to change shape, fold and deform shape, place flat on a surface, against wall, and hang it on a hook
books	place flat on surfaces, open and close books, and place horizontally with cover up and vertically
bottles	open and close lid, fill different types of liquids, insert/take out drinking straws
bowls	place normally and up-side down, place objects/food inside, open and close lid (if bowl has lid)
clothes	put on a hanger, place on floor in a heap, fold properly, hang outside for drying, place randomly on different surfaces
cups	place normally and up-side down, fill different types of liquid, place on coaster
decorations	remove and change the arrangement of detachable items (if possible), deform flexible items, rearrange, switch on/off lights (for those that have light fixtures), place in different stable poses, capture from viewpoints with significant change in appearance, place in complicated backgrounds, and different lighting conditions (especially ones that have reflective surfaces/have light fixtures)
headphones	change orientations of microphone and speakers, change how the wire is placed, lie flat on table, hanging and in an upright position
telephones	change receiver position and orientation, place hanging on wall and flat on table, change shape of wire, lights on/off while operating
pillows	deform and place on different supporting surfaces such as floor and against wall, put force on pillow to deform in several ways, fold (if possible)
plants	move pots to different locations (only if not fixed), different lighting conditions (sunlight/cloudy/night), with and without flowers (when applicable), place twigs in different orientations, trim leaves/twigs
plates	place normally and up-side down, place objects/food on surface
remotes	pressing buttons/joysticks, open battery compartment, removing batteries, lights on/off upon pressing buttons
retail products	Remove detachable components and rearrange by placing in different poses, opening cap/lid and placing it beside the object (when applicable), deform objects (when applicable), place on all possible object surfaces
toys	removing detachable components (when applicable), place detachable components in various orientations, deform soft objects while placing them in different poses, decorate toys with accessories
ties	tie a knot and place normally and in an arbitrary pose, open knot and place normally and in arbitrary pose, roll the tie, hang on a hook
towels	fold and place on a surface, fold and place on a hanger, put on a surface in a heap (with different types of deformations), place on arbitrary-shaped surfaces, tie knots
trolley bags	place in different stable poses (upright on wheels/flat on ground, on different surfaces along length/breadth), with the handle inside/drawn out, open the bag and place it on the floor and in an upright condition, place things inside, rearrange items and empty the bag
tumblers	place normally and upside down, fill different types of liquids, put complicated backgrounds (especially for transparent ones), place in different lighting conditions (especially for those with reflective surfaces)
umbrellas	open state, closed state with/without tying, fold (for compact foldable umbrellas), handle inside/drawn out, different backgrounds (inside/outside), folded and hanging
vegetables	raw state, arrange multiple items in different ways, place on different utensils, peel skin, chop into large chunks, chop finely

Table 6. This table mentions possible state changes of the objects from 21 categories that are considered during the data collection. These include *diverse* types of changes arising from *everyday interactions with objects*. For each state, several images of the object were captured from different transformations such as pose, light, viewpoint changes etc.

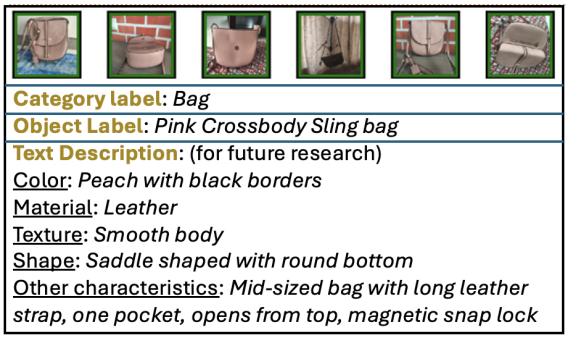

Figure 11. shows the images of an object in various states and poses captured from arbitrary viewpoints, category label, object label, and text description.

them to recognize objects despite state changes. This can be achieved by presenting pairs of images of the same object in different states and prompting the model either to identify the object or to elaborate on the differences between the two states. To facilitate this, we can derive an *expanded dataset from OWSC by generating over 400k pairwise image combinations and annotating them with variations in state, pose, background, and lighting*, which we plan to pursue as future work.

5. The dataset can support few-shot learning via pair and triplet mining in deep metric learning, enabling models to learn discriminative embeddings from limited data by leveraging state and transformation variations for better generalization.

Challenges: The following aspects make our OWSC dataset challenging:

1. Significant appearance changes due to *state changes* and other transformations such as *arbitrary pose, viewpoint*, lighting, and background changes
2. Presence of several similar-looking objects with *fine-grained differences* which makes recognition and retrieval of the correct object-identity challenging especially when there are state changes.
3. Images of the objects are captured *in the wild* with clut-

Method	Classification (Accuracy %)					Retrieval (mAP %)				
	Category		Object		Avg.	Category		Object		Avg.
	SV	MV	SV	MV		SV	MV	SV	MV	
PI-CNN [3]	51.63	68.00	26.64	40.00	46.57	41.51	58.18	23.76	55.62	44.76
PI-Proxy [3]	48.03	69.33	26.93	42.67	46.74	39.35	54.85	24.42	58.80	44.36
PI-TC [3]	61.06	80.00	37.09	53.33	57.87	48.08	**69.91**	30.31	68.68	54.25
PiRO [11]	61.74	84.00	51.97	78.67	69.10	46.72	47.13	46.57	84.25	56.17
Ours	**70.20**	**90.67**	**58.68**	**85.33**	**76.22**	**52.46**	53.51	**51.27**	**90.24**	**61.87**

Table 7. Comparison of classification and retrieval performance for the category and object-based tasks against several state-of-the-art pose-invariant methods on novel (unseen) objects using the OWSC-GN split. SV and MV denote either a single image or multiple images used at the time of inference respectively.

tered and complex backgrounds which facilitates evaluation of algorithms in real-world scenarios.

8. Generalization to Novel Objects

In the main paper, we evaluated the model's robustness in recognizing and retrieving images of the same object (seen during training) across various transformations and state changes. To further assess the *generalization of the learned representations to novel objects*, we introduced the OWSC-GN split, which contains 2,509 images of 75 objects not included in the OWSC-SI training set. For each object, images from one state were randomly assigned to the gallery set, while images from the remaining states were allocated to the probe set, as shown in Fig. 10.

8.1. Benchmarking on the OWSC-GN split

We assess the generalizability of the representations learned by a model trained on OWSC-SI, by evaluating it on the OWSC-GN split comprising unseen objects. The evaluation involves matching the probe images of unseen objects (under various state changes and other transformations) to their corresponding gallery images (in a different state than probe images) via similarity search. This setup measures the model's ability to handle novel objects under state variations and other transformations. The recognition and retrieval tasks follow the same protocol, as described in Section 3(D). We report the results in Table 7.

Overall, we observe an average improvement of 7.1% in classification and 5.7% in retrieval tasks, with more substantial gains for object-level tasks. Our method consistently outperforms prior approaches on category-level classification, object-level classification, and object-level retrieval tasks. Interestingly, PI-TC achieves the best performance on category-level retrieval, likely because it tightly clusters all the images of the objects within the same category, though it struggles to distinguish between individual objects. In contrast, our method is better at discriminating between different objects, as reflected in the superior object-level retrieval performance.

Training Dataset	Sampling Method	Classification (Accuracy %)					Retrieval (mAP %)				
		Category		Object		Avg.	Category		Object		Avg.
		SV	MV	SV	MV		SV	MV	SV	MV	
Object-PI	Same category	51.2	78.7	37.7	69.3	59.2	37.5	55.1	32.1	77.9	50.7
	Curriculum (Ours)	**55.1**	**82.7**	**43.2**	**74.7**	**63.9**	**41.1**	**57.3**	**36.7**	**80.5**	**53.9**
OWSC-SI	Same category	61.7	84.0	52.0	78.7	69.1	46.7	47.1	46.6	84.3	56.2
	Curriculum (Ours)	**70.2**	**90.7**	**58.7**	**85.3**	**76.2**	**52.5**	**53.5**	**51.3**	**90.2**	**61.9**

Table 8. Results of cross-dataset performance evaluation. Model is trained on different datasets such as ObjectPI (capturing pose variations) or OWSC-SI (additionally capturing state variations) and tested on novel objects in the OWSC-GN split (capturing state changes and other transformations such as viewpoint, pose, background, and lighting changes).

8.2. Using OWSC-GN for Cross-Dataset Evaluation

Cross-dataset evaluation involves training a model on one dataset and testing it on another, providing a measure of generalization and robustness to conditions not seen during training. In this context, the *OWSC-GN split serves as an evaluation benchmark to test the robustness of learned representations to state changes and other real-world transformations*, including variations in viewpoint, pose, background, and lighting.

To illustrate this, we evaluate the PiRO model [11] and our method trained on both the ObjectPI [3] dataset and the OWSC-SI split of our ObjectsWithStateChange dataset, testing them on the OWSC-GN split containing unseen objects. The results in Table 8 show:

1. Models trained on OWSC-SI outperform those trained on ObjectPI, as OWSC-SI captures state variations in addition to the pose and transformation variations present in ObjectPI. This setup can be used to assess how well models generalize to state changes when trained on different datasets.
2. Using object pairs mined via curriculum learning consistently improves performance over randomly sampling object pairs from the same category as in [11], across all tasks and datasets.

Training Ours with State Change	Classification (Accuracy %)					Retrieval (mAP %)				
	Category		Object		Avg.	Category		Object		Avg.
	SV	MV	SV	MV		SV	MV	SV	MV	
No	57.55	84.00	47.25	74.67	65.87	42.61	50.09	41.50	80.30	53.63
Yes	**70.20**	**90.67**	**58.68**	**85.33**	**76.22**	**52.46**	**53.51**	**51.27**	**90.24**	**61.87**
Improvement	12.65	6.67	11.43	10.66	10.35	9.85	3.42	9.97	9.94	8.24

Table 9. Comparison of classification and retrieval performance on both category- and object-level tasks for our method trained *with and without state changes*. Results on the OWSC-GN split show that incorporating state variations leads to substantial performance improvements.

9. Further Ablation Studies

9.1. Ablation With and Without State Changes

This ablation study investigates the impact of learning from data with and without state changes. To this end, we train our method under two settings:

• **Without state changes**: This training set includes images of objects in OWSC-SI in a single state (but with other transformations).

• **With state changes**: The training set incorporates multiple state variations for each object in OWSC-SI (along with other transformations).

We then evaluate both models, trained with our curriculum learning approach, on the OWSC-GN dataset containing unseen objects amidst state variations and other transformations, and present the results in Table 9. The model trained with state variations shows a significant performance improvement in recognition and retrieval tasks at both the category and object levels, highlighting the value of learning from datasets that reflect object appearance changes across states. A similar conclusion is supported by Tbl. 8, which compares our method's performance when trained on ObjectPI (capturing only pose variations) versus OWSC-SI (which additionally includes state variations).

9.2. Detailed Ablation of Curriculum Learning

9.2.1. Evaluation on different datasets

PiRO uses multi-view images of a pair of objects randomly sampled from the same category, whereas our new method additionally samples images of a pair of neighboring objects from the same category and other categories. This ablation answers the question whether the additional computational burden entailed by this sampling (ref. Supplement Sec. 10) is worth the effort.

We compare the two sampling strategies using our OWSC dataset on state-invariant recognition and retrieval tasks (Tbl. 10a), as well as on pose-invariant recognition and retrieval tasks with three publicly available multi-view datasets—ObjectPI [3], ModelNet-40 [21], and FG3D [9] (Tbl. 10b). The results show that our curriculum learning strategy improves average recognition and retrieval performance across all datasets, with more substantial gains on object-based tasks. Notably, in Tbl. 10a, curriculum learning improves single-view object recognition by 7.9% and retrieval by 9.2% for OWSC-SI split, and improves single-view object recognition by 6.7% and retrieval by 4.7% for OWSC-GN split, compared to sampling objects randomly from the same category [11]. Similarly, in Tbl. 10b, we observe that our sampling strategy improves single-view object recognition and retrieval performance on the multi-view datasets (ObjectPI, ModelNet, and FG3D) for pose-invariant tasks. For multi-view object recognition and retrieval tasks, we see a significant improvement on our OWSC dataset

Dataset	Sampling Method	Classification (Accuracy %)					Retrieval (mAP %)				
		Category		Object		Avg.	Category		Object		Avg.
		SV	MV	SV	MV		SV	MV	SV	MV	
OWSC-SI	Same category	**87.1**	**91.2**	68.7	76.4	80.9	88.5	**93.2**	68.8	82.9	83.4
	Curriculum (Ours)	86.9	89.4	**76.6**	**83.4**	**84.1**	**88.6**	92.4	**78.0**	**88.0**	**86.8**
OWSC-GN	Same category	61.7	84.0	52.0	78.7	69.1	46.7	47.1	46.6	84.3	56.2
	Curriculum (Ours)	**70.2**	**90.7**	**58.7**	**85.3**	**76.2**	**52.5**	**53.5**	**51.3**	**90.2**	**61.9**

(a) Results on single and multi-image state-invariant recognition and retrieval tasks on our ObjectsWithStateChange (OWSC) dataset.

Dataset	Sampling Method	Classification (Accuracy %)					Retrieval (mAP %)				
		Category		Object		Avg.	Category		Object		Avg.
		SV	MV	SV	MV		SV	MV	SV	MV	
Object PI	Same category	**71.3**	83.7	92.7	98.0	86.4	**65.7**	**83.4**	81.0	99.0	82.3
	Curriculum (Ours)	70.3	**85.7**	**96.2**	**99.0**	**87.8**	64.9	82.4	**86.3**	**99.3**	**83.2**
Model Net40	Same category	**85.2**	88.9	94.0	96.9	91.2	**79.7**	**86.1**	84.0	98.2	87.0
	Curriculum (Ours)	84.7	**89.0**	**95.5**	**97.5**	**91.7**	79.6	85.8	**87.9**	**98.6**	**88.0**
FG3D	Same category	**79.0**	**81.8**	83.1	91.5	83.9	**68.1**	**74.4**	73.0	95.5	77.8
	Curriculum (Ours)	78.3	81.3	**84.7**	**92.7**	**84.3**	67.6	73.7	**77.3**	**96.1**	**78.7**

(b) Results on single and multi-view pose-invariant recognition and retrieval tasks on three publicly available multi-view datasets.

Table 10. Ablation study to understand the effect of sampling training examples. PiRO [11] randomly sampled object pairs from the same category, whereas our method additionally samples a pair of neighboring objects from the same and other categories. We observe that our curriculum learning strategy improves recognition and retrieval performance on object-level tasks.

(both splits) in Tbl. 10a, while improvements on ObjectPI, ModelNet-40, and FG3D are smaller but still consistent in Tbl. 10b. In the following subsection, we provide a more detailed analysis of these performance improvements.

9.2.2. Discussion about performance improvements

(A) Greater performance gains on OWSC compared to existing multi-view object datasets: As shown in Fig. 5, state-induced variations in OWSC introduce both substantial intra-object class variance and often cause distinct objects, both within and across categories, to appear visually similar (e.g., a folded or crumpled blanket, clothing item, or towel may look alike). Moreover, OWSC contains fine-grained object classes with inherently high inter-object similarity. This is validated by OWSC having the lowest initial minimum inter-object distances across all object classes compared to ObjectPI, ModelNet, and FG3D, highlighting its stronger inter-object similarity. Thus, such challenging visually similar instances of different objects are more prevalent in OWSC than existing multi-view object datasets that primarily capture pose variations. Our curriculum learning strategy is especially effective at separating such similar objects in the learned embedding space, thereby leading to greater performance gains on OWSC as compared to existing multi-view object datasets that primarily capture

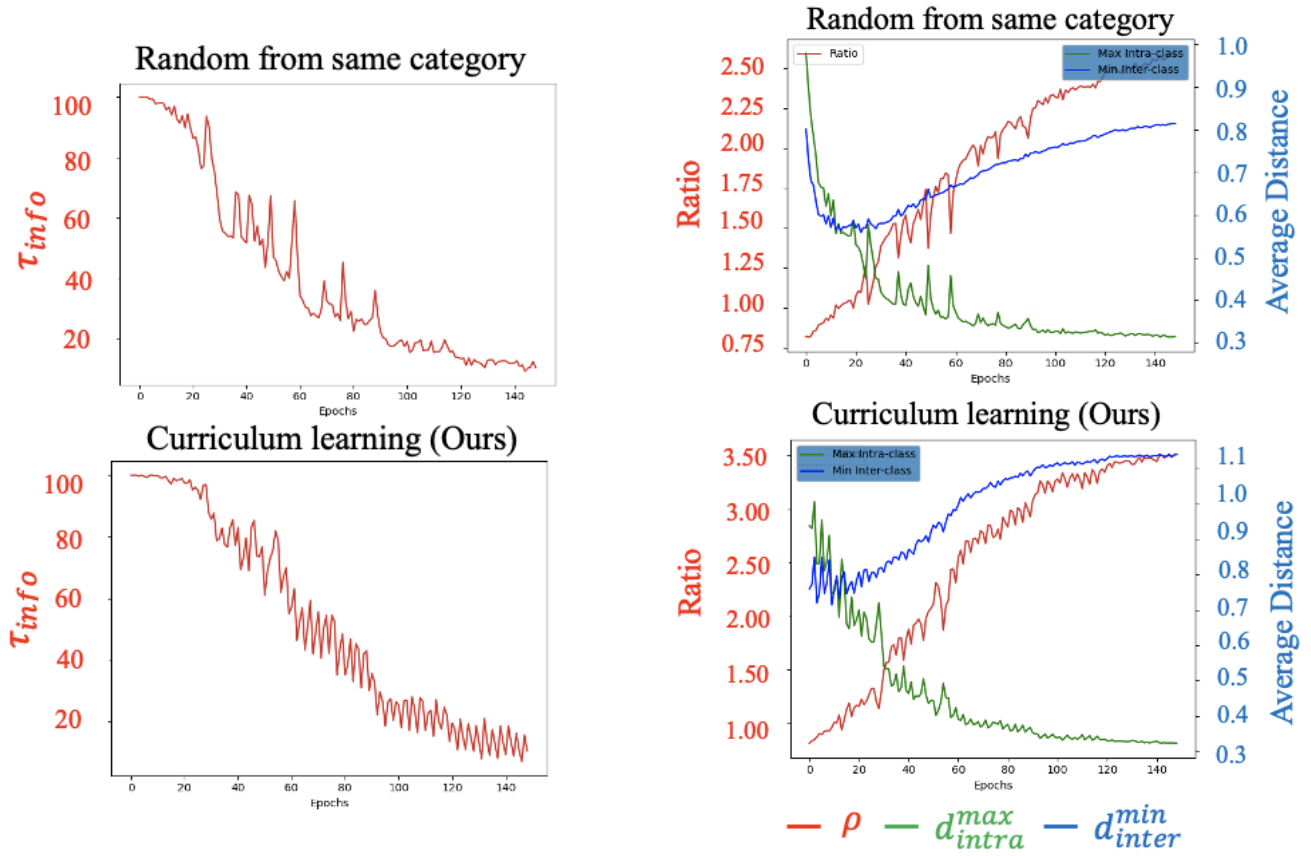

Figure 12. The figure on the left shows the percentage of informative examples (τ_{info}) selected using different sampling strategies vs epochs during training on our OWSC dataset. Our curriculum learning based mining yields more informative samples, especially during the initial stages of the training, as compared to random sampling. The figure on the right shows the variation of inter-object (d_{inter}^{min}), intra-object (d_{intra}^{max}) distances, and their ratio ρ vs epochs during training for different sampling strategies. Our curriculum learning approach clusters the samples from distinct objects better (as indicated by the higher inter-object distances in blue lines and higher ρ values indicated in red).

pose variations.

(B) Performance improvements are greater for object-level tasks than category-level tasks: As shown in Tbls. 2, 5, and 10, we observe larger performance gains on object-level tasks compared to category-level tasks. This is because our curriculum learning strategy prioritizes distinguishing visually similar objects to enhance performance on object-level tasks rather than category-level tasks. Specifically, our approach samples images from a pair of objects and optimizes their distances in both the object and category embedding spaces simultaneously. In the object space, informative samples are similar object pairs that lie close together, since our goal is to separate different objects well in the object space during training. Conversely, in the category space, informative samples are dissimilar object pairs from the same category that are farther apart, since our goal is to cluster them together in the category space during training. During stages S2 and S3 of our curriculum learning approach in Sec. 4.3, we explicitly select similar object pairs with lower inter-object distances, as training progresses. While these pairs are difficult to separate in the object space, they are relatively easy to cluster in the category space due to their highly similar attributes, making them not as informative for learning category-specific features. As a result, our method yields higher performance improvements on object-level tasks than on category-level tasks.

9.3. Optimization of Intra-Object and Inter-Object Distances During Training:

In this section, we discuss the ablation study related to optimization of the intra-object and inter-object distances presented in Sec. 5.2(C), in more detail. Specifically, we compare how these distances are optimized and the informativeness of the samples when mining object pairs using our proposed curriculum learning approach as compared to randomly sampling object pairs in [11] during training.

In Fig. 12 (left), we plot the percentage of training samples that are informative (τ_{info} in red). A training sample

is considered informative if the loss L_{piobj} for that sample is non-zero. Being a margin-based loss, if the training samples of different objects are already well separated or if the training samples of the same object are already well clustered then the loss L_{piobj} for that sample will be zero and the sample is uninformative. If we compare τ_{info} at any given epoch, the curriculum learning approach generates more informative samples than sampling objects from the same category, especially during the initial stages of training.

In Fig. 12 (right), we plot the maximum intra-object distance (d^{max}_{intra} in green), the minimum inter-object distance between object-identity classes (d^{min}_{inter} in blue), and the ratio $\rho = \frac{d^{min}_{inter}}{d^{max}_{intra}}$ (in red) during training, similar to [11]. These metrics are used to monitor the compactness and separability of object-identity classes. Specifically, a lower d^{max}_{intra}, and higher d^{min}_{inter} and ρ values would indicate that the embeddings of the same object-identity class are well clustered and are well separated from the embeddings of other object-identity classes.

We observe that both the sampling methods decrease d^{max}_{intra} to the same extent. However, the curriculum learning strategy increases d^{min}_{inter} and thereby increases ρ to a greater extent than random sampling. Therefore, the object-identity classes are better separated (indicated by the blue lines), and this results in improved performance on object-level tasks.

10. Time Complexity Analysis

For this analysis, we assume there are N^O objects distributed across N^C categories in the dataset. The distribution of objects per category is assumed to be uniform, with each category containing an average of $N^O_C = \frac{N^O}{N^C}$ objects. Our goal is to examine how the proposed sampling strategy in Sec. 4.3 scales with the number of categories and objects.

It is important to note that to reduce the complexity of determining neighboring objects in the embedding space, we use a single aggregated embedding for each object extracted from our encoder, regardless of the number of images associated with it. So, the number of images per object is not included in this analysis.

10.1. Analysis of the Proposed Sampling Strategy

At the beginning of each epoch, our algorithm utilizes either of the three distinct sampling strategies:

(S1) For each object, we randomly select another object from the same category. This operation has a time complexity of $\mathcal{O}(1)$. When applied to the entire dataset, the overall time complexity is $\mathcal{O}(N^O)$.

(S2) For each object, we construct a list of similar objects within the same category and randomly sample an object pair from this list. This requires finding approximate all-nearest-neighbors [16] for all objects in the same category, which incurs a complexity of $\mathcal{O}(N^O_C \log N^O_C)$ for each category using efficient ANN techniques [4]. Consequently, the total time complexity for the entire dataset is $\mathcal{O}(N^C N^O_C \log N^O_C) = \mathcal{O}(N^O \log N^O_C)$.

(S3) This strategy involves partitioning the embedding space using k-means clustering and building an inverted index, which has a time complexity of $\mathcal{O}(KN^OI)$, where K representing the number of clusters and I denoting the number of iterations are constants. For each object, we randomly sample another object from the same partition, which has a time complexity of $\mathcal{O}(1)$. Thus, for the entire dataset, the overall time complexity for partitioning and sampling object pairs is $\mathcal{O}(KN^OI) \approx \mathcal{O}(N^O)$, since both K and I are constants independent of N^O or N^C.

Overall, the average time complexity of the algorithm is dominated by the second sampling strategy, yielding $\mathcal{O}(N^O \log N^O_C)$ for the entire dataset comprising N^O objects. *For each object, the time complexity of finding another object for comparison is therefore $\mathcal{O}(\log N^O_C)$, which is sub-linear with respect to the number of objects per category in the dataset.*

10.2. Comparison with Other Methods:

In this section, we compare the time complexity *per object* for different methods. The state-of-the-art method, PiRO [11], randomly selects another object from the same category for each object, resulting in a time complexity of $\mathcal{O}(1)$ per object. In contrast, prior work (PI-TC) [3] compares each object with the *nearest* multi-view embedding of the object and the *nearest* proxy embedding for the categories, which incurs a complexity of $\mathcal{O}(N^C + N^O) = \mathcal{O}(N^O)$ per object, given that $N^C < N^O$.

Our method employs different sampling strategies across various epochs. Sampling strategies S1 and S3 share the same complexity as PiRO [11], assuming that K and I remain small relative to N^O. However, S2 exhibits a higher complexity than PiRO due to the additional $log N^O_C$ term. For most datasets, N^O_C is relatively small, so the logarithmic term does not significantly impact overall complexity. Nevertheless, for datasets with a large number of objects per category, sampling strategy S2 would have higher complexity than PiRO, although it will still have better complexity than PI-TC [3].

11. Implementation Details:

For a fair comparison with prior methods [3, 11], VGG-16 is used as the CNN backbone. The last FC layers are modified to generate 2048-D embeddings and are initialized with random weights. Two-layer single-head self-attention layers are used for each embedding space with a dropout of 0.25. Training involves jointly optimizing the category and object-based losses, as described in Eqs. 1, 2. Margins $\alpha = 0.25$ and $\beta = 1.00$ are set for the object space, while

Figure 13. This figure is a zoomed in version of Fig 8 in the main paper that shows the results of single-image object retrieval from our dataset using our proposed method. The blue bounding box highlights the query image, while the green bounding box indicates the successful retrieval of an image of the correct object-identity. Our method demonstrates accurate retrieval of images of the same object under different transformations in (a) on the left side and effectively distinguishes the correct object from similar-looking ones as shown in consecutive rows in (b) on the right side.

margins $\theta = 0.25$ and $\gamma = 4.00$ are set for the category space, which are the same as [11]. V is set to 12 images per object, and the images are resized to 224×224 and normalized. We use the Adam optimizer with a learning rate of $5e^{-5}$. We train for 150 epochs and use the step scheduler that reduces the learning rate by half after every 30 epochs. For curriculum learning, at the beginning of each epoch an IVFFlat index is built using FAISS [4] using the currently learned multi-image object embeddings (with one aggregated embedding per object). For this, the quantizer used is FlatL2 and the object embedding space is divided into $f(N_e) = max(min(cN_e, n_{max}), n_{min})$ partitions, where N_e is the current training epoch, c=2, n_{min}=8 and n_{max}=100.

12. Qualitative Fine-grained Retrieval Results

As mentioned earlier in Section 3, our dataset comprises visually similar objects from each of the 21 categories. In this section, we show some qualitative single-image retrieval results of visually similar objects from each category in our dataset in Figs. 14 and 15.

In these figures, the results are presented in two columns that show the results for similar-looking objects from the same category in each row. Firstly, this illustrates that our dataset has several similar-looking objects from each category with very subtle differences in appearance and hence can facilitate research in fine-grained object retrieval tasks. Secondly, the results indicate that our algorithm is able to retrieve images of the same object-identity with high mAP

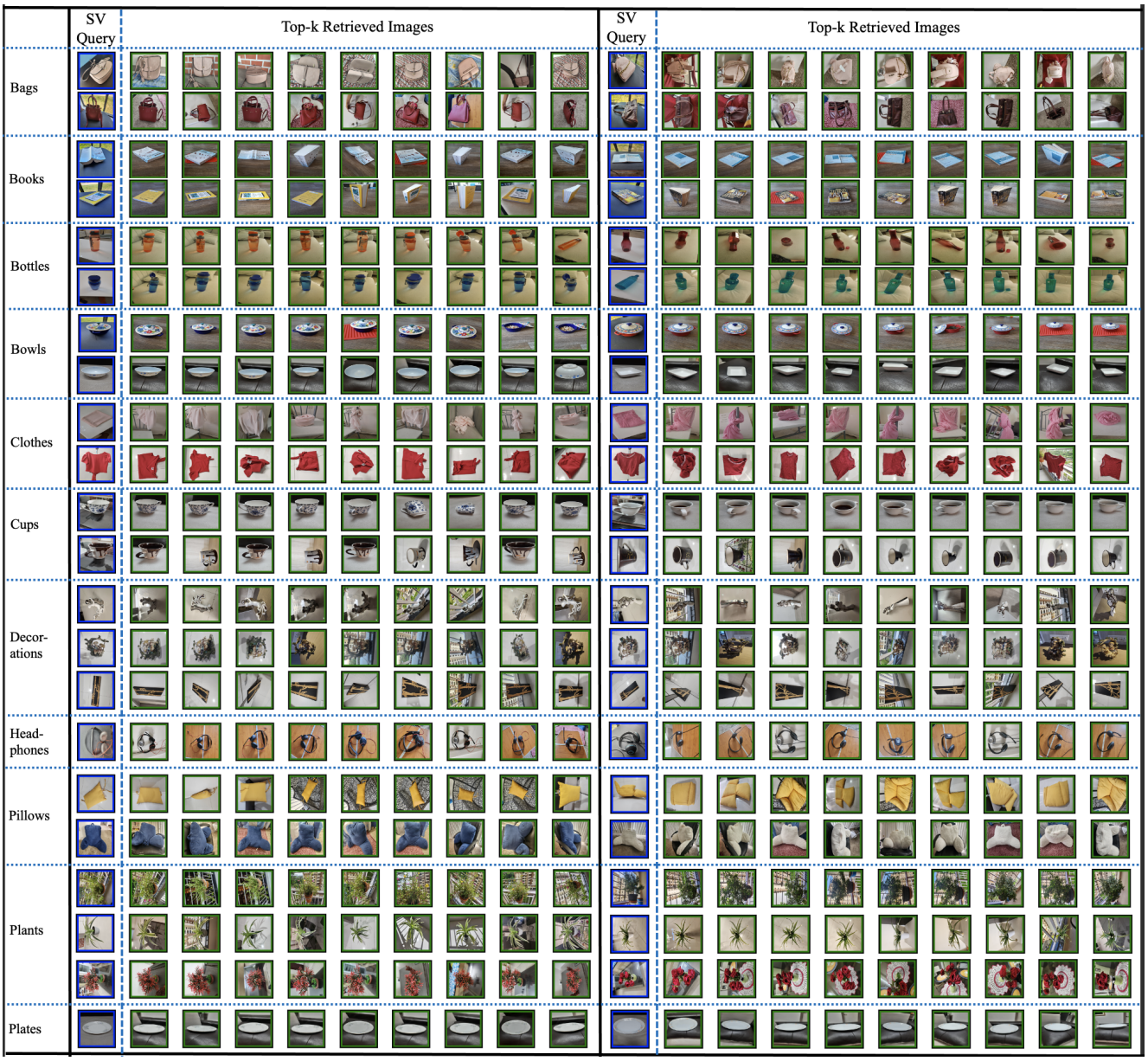

Figure 14. This figure shows single-view object retrieval results of the objects from the first 11 categories. The two columns separated by the thick black line in the middle show results for similar-looking objects within each category. Specifically, in the left column, we show the top-k retrieval results for a SV query image while in the same row of the right column, we show the SV retrieval results for another similar object from the same category. The figure is best viewed when zoomed in. The blue bounding boxes around the images indicate the SV query image while the green and red bounding boxes indicate the correctly and incorrectly retrieved images of the same object-identity as the query image respectively. The results for the other categories are shown on the next page.

despite having similar-looking objects from each category and different state changes and other transformations from arbitrary viewpoints.

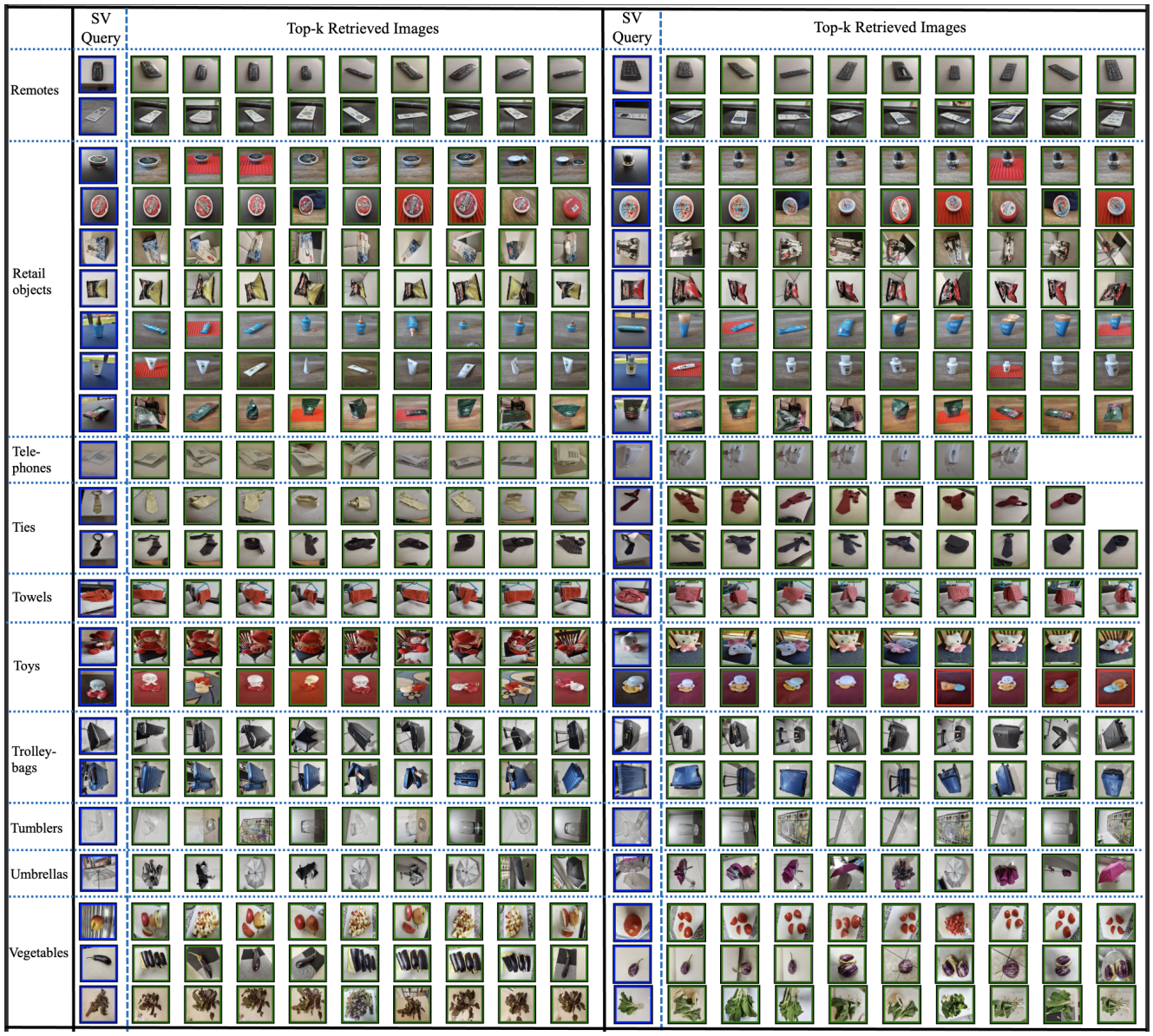

Figure 15. This figure shows single-view object retrieval results of the objects from the remaining 10 categories. Following the same format as the previous figure, the two columns separated by the thick black line in the middle show results for similar-looking objects within each category. Specifically, in the left column, we show the top-k retrieval results for a SV query image while in the same row of the right column, we show the SV retrieval results for another similar object from the same category. The figure is best viewed when zoomed in. The blue bounding boxes around the images indicate the SV query image while the green and red bounding boxes indicate the correctly and incorrectly retrieved images of the same object-identity as the query image respectively.