Dual Pose-invariant Embeddings: Learning Category and Object-specific Discriminative Representations for Recognition and Retrieval

Supplementary Material

In the supplementary material, we present additional results that could not be reported in detail in the main paper due to space constraints. Our supplemental is organized as follows. In Sec. 6, we present UMAP visualizations of the learned pose-invariant embeddings for the ObjectPI, ModelNet40, and FG3D datasets. In Sec. 7, we present a detailed ablation study of the different components of the proposed pose-invariant object loss. We investigate how the inter-class and intra-class distances for the object-identity classes are optimized in the object embedding space, and further explain how the separation of object-identity classes leads to significant performance improvement on object-level tasks. Furthermore, we investigate how the object-identity classes are better separated when learning dual embedding spaces as compared to a single embedding space in Sec. 8. Subsequently in Sec. 9, we study the effect of embedding dimensionality on category and object-based classification and retrieval tasks. Next, we illustrate how self-attention captures correlations between different views of an object using multi-view attention maps in Sec. 10, and finally present qualitative single-view object retrieval results in Sec. 11. At the end, additional details of the category and object-level tasks are provided in Sec. 12.

6. UMAP Visualization of Pose-invariant Embeddings

For a qualitative understanding of the effectiveness of our approach, we compare the embeddings generated by the prior pose-invariant methods (specifically, PI-CNN, PI-Proxy, and PI-TC) in [7] with our method. For this, we use UMAP to project the embeddings into the 2-dimensional space for visualization. In Fig. 8, we compare the UMAP plots for a subset of the test dataset of ModelNet-40. Since ModelNet-40 has a large number of objects in the test dataset, we choose 100 objects for visualization from five mutually confusing categories such as tables and desks, chairs, stools, and sofas. In the category plots, we use five distinct colors to indicate instances from each of the five categories. In the object plots, instances of each object-identity

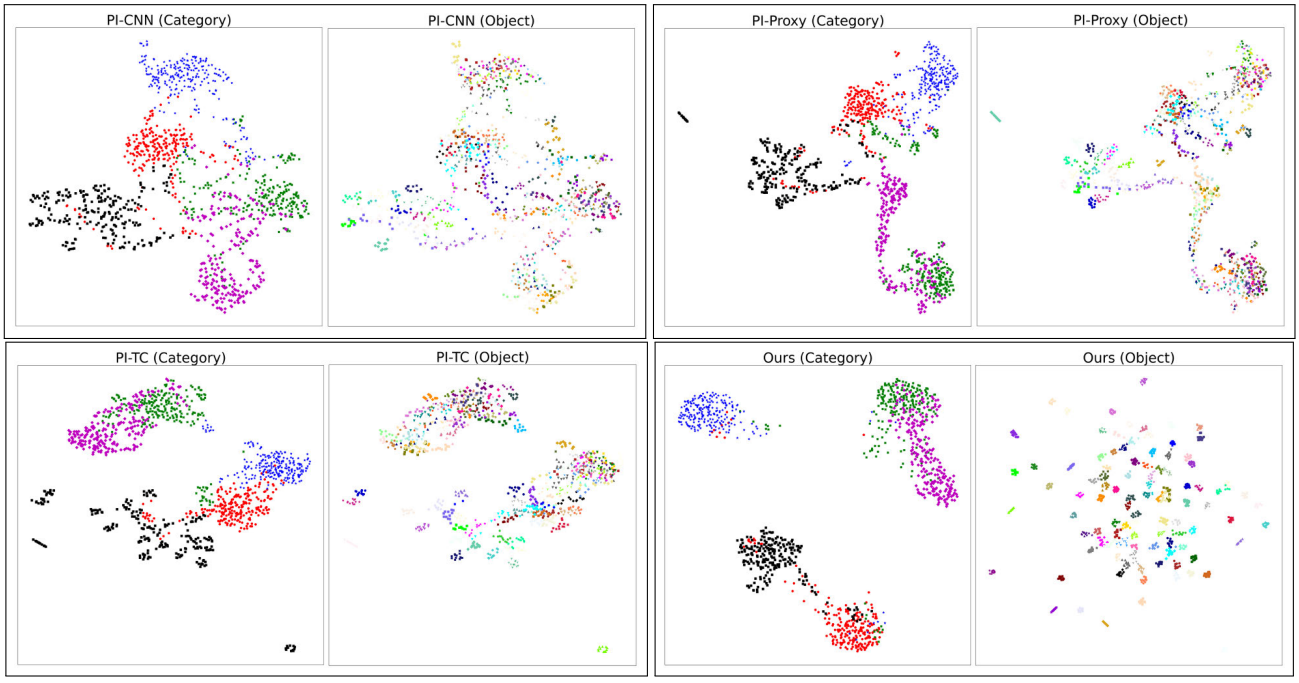


Figure 8. *Qualitative comparison of the embedding space learned for a subset of the ModelNet40 test dataset (from 5 categories such as table, desk, chair, stool, sofa with 100 objects) by prior pose-invariant methods [7] and our method (bottom-right). In the category plots (to the left of each subfigure), we use five distinct colors to indicate instances from each of the categories. In the object plots (to the right of each subfigure), instances of the same object-identity class are indicated by a unique color and shape.*

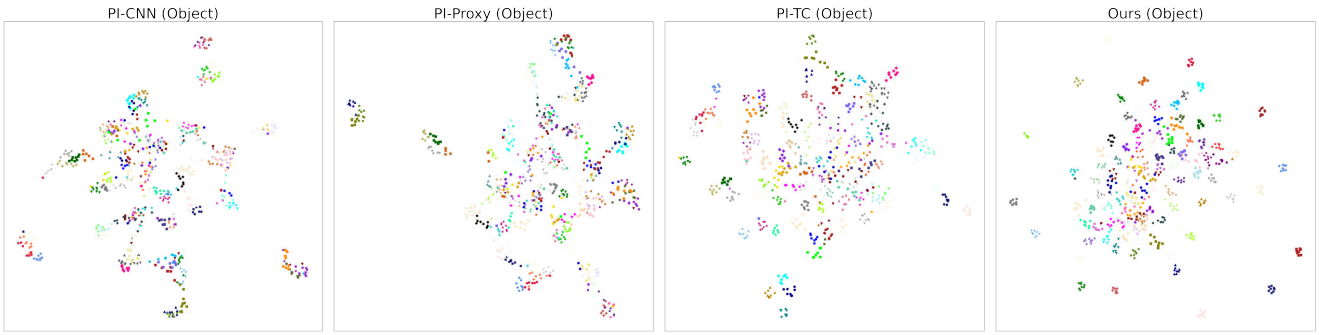
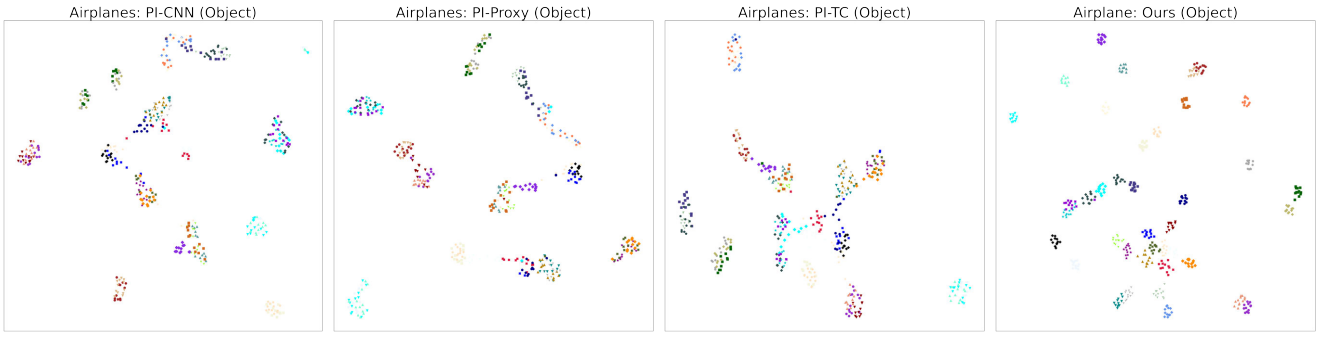
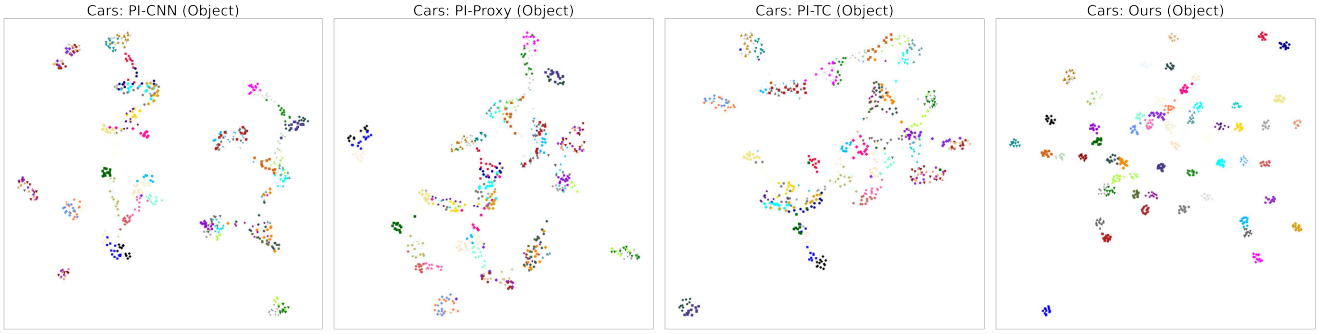


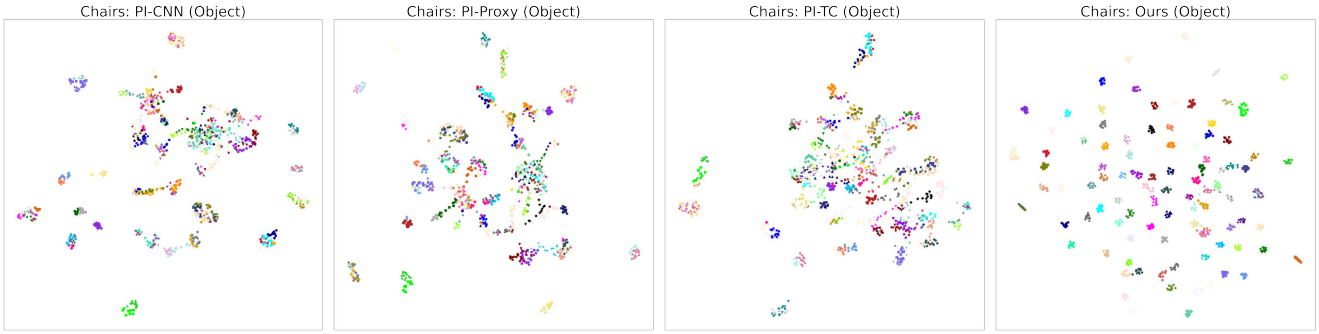
Figure 9. *Comparison of the object embedding space learned for the ObjectPI test dataset (with 98 objects) by prior pose-invariant methods [7] and our method (right). Each instance is an object view and each object-identity class is denoted by a unique color and shape.*



(a) Object-identity embeddings for 39 airplane objects (3 objects each from 13 airplane categories)



(b) Object-identity embeddings for 60 car objects (3 objects each from 20 car categories)



(c) Object-identity embeddings for 99 chair objects (3 objects each from 33 chair categories)

Figure 10. *Comparison of the object embedding space learned for the FG3D test dataset (comprising objects with fine-grained differences from 13 airplane categories in (a), 20 car categories in (b), and 33 chair categories in (c)) by prior pose-invariant methods [7] and our method (right). Each instance is an object view and each object-identity class is denoted by a unique color and shape. It can be observed that our object-identity embeddings are better clustered and separated from other objects as compared to prior methods.*

class are indicated by a unique color and shape.

Prior pose-invariant methods (PI-CNN, PI-Proxy, and PI-TC) in [7] learn a single embedding space. For each of these methods, we show the UMAP visualizations of the same embedding space with the category and object-identity labels in the two subfigures (titled category and object). As mentioned in the paper, prior work focused primarily on learning category-specific embeddings, with the object-to-object variations within each category represented by the variations in the embedding vectors within the same embedding space. Specifically, we observe that for PI-CNN and PI-Proxy (in the top row of Fig. 8), the pose-invariant embeddings for object-identity classes belonging to the same category are not well-separated leading to poor performance on object-based tasks reported in the main paper in Tables 2, 3. PI-TC (bottom-left of Fig. 8) separates embeddings of the nearest neighbor object-identity classes in the embedding space leading to comparatively better performance.

In contrast, our method decouples the category and object representations in separate embedding spaces leading to a better separation of both the category and object-identity embeddings, as can be seen in the bottom-right of Fig. 8. The most notable difference with prior state-of-the-art is in regards to the learnt object-identity embeddings. Hence, for the other datasets we compare the object-identity embeddings generated by our method and prior pose-invariant methods. In Fig. 9, we visualize the embeddings for the ObjectPI test dataset comprising 98 objects from 25 categories. The FG3D dataset has 66 fine-grained categories that comprise 13 types of airplanes, 20 types of cars, and 33 types of chairs. We sample 3 objects per category and show the object-identity embeddings for the airplane, car, and chair objects separately in Fig. 10 (a), (b), and (c) respectively. For all the datasets, we observe that the object-identity classes are better clustered and separated for our method as compared to prior methods.

We conjecture that our method better separates the object-identity classes for two reasons. First, our method separates confusing instances of objects from the same category that would otherwise be much too close together in the embedding space, as we will explain in detail in Section 7. Second, our method captures category and object-specific discriminative features in separate embedding spaces. Intuitively, this allows us to simultaneously capture common attributes between objects from the same category in the category embedding space and discriminative features to distinguish between them in the object embedding space, as opposed to learning representations to satisfy these conflicting objectives in the same embedding space. This strategy leads to better separability of the object-identity embeddings when learning a dual space as compared to learning a single space, as we will explain in detail in Section 8.

7. Ablation of Pose-invariant Object Loss

As explained in Sec. 2(B) of the main paper, prior approaches primarily focus on clustering the single-view embeddings of each object-identity class close to their multi-view embeddings but do not effectively separate embeddings from different object-identity classes. To ameliorate this, our proposed pose-invariant object loss is designed to separate different object-identity classes, and in this section, we investigate its importance for good performance on object-based tasks.

Our pose-invariant object loss in Eqn. 8 has two components – clustering loss ($\mathcal{L}_{intra}$) that reduces the intra-object distances by clustering different views of the same object-identity class, similar to prior approaches. Additionally, we add a separation loss ($\mathcal{L}_{inter}$) that increases the inter-object distances by separating confusing instances of different object-identity classes from the same category. To understand the effectiveness of each component, we train

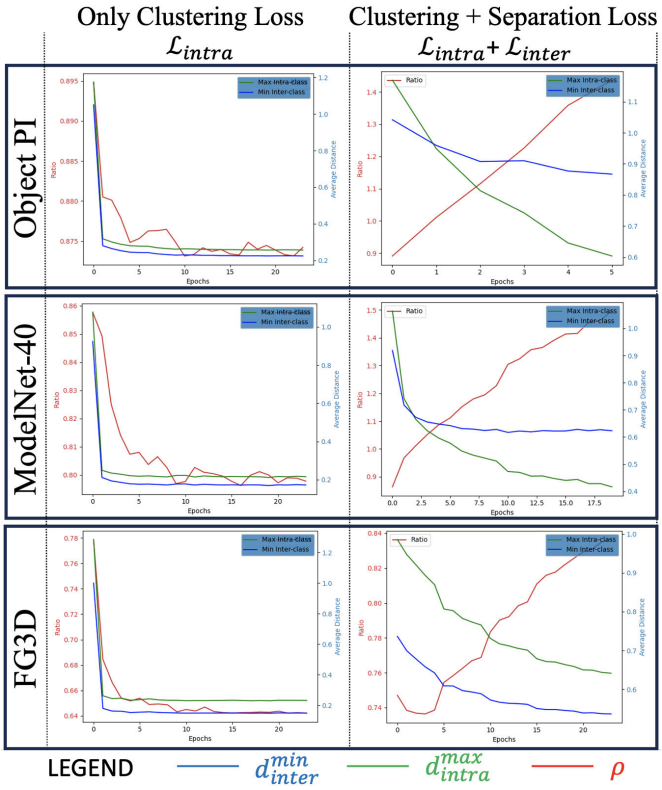


Figure 11. *This figure illustrates how the minimum inter-class distances between object-identity classes (d^{min}_{inter} in blue), maximum intra-class distances (d^{max}_{intra} in green), and the ratio of the two distances (ρ in red) change as the different components of our pose-invariant object loss are optimized for all three datasets. When training using the clustering loss only, ρ decreases. Whereas, when training using both the losses together, ρ increases indicating better compactness and separability of object-identity classes.*

Datasets	Losses	Optimized distances during training			Test Performance on Object-level tasks			
					Classification (Acc. %)		Retrieval (mAP %)	
		$d^{max}_{intra}(\downarrow)$	$d^{min}_{inter}(\uparrow)$	$\rho(\uparrow)$	Single-view	Multi-view	Single-view	Multi-view
ObjectPI	$\mathcal{L}_{intra}$	0.26	0.23	0.87	84.9	87.8	59.8	93.2
	$\mathcal{L}_{intra} + \mathcal{L}_{inter}$	0.60	0.90	**1.50**	**92.7**	**98.0**	**81.0**	**99.0**
ModelNet-40	$\mathcal{L}_{intra}$	0.22	0.18	0.80	68.5	68.6	43.0	76.2
	$\mathcal{L}_{intra} + \mathcal{L}_{inter}$	0.41	0.62	**1.51**	**93.7**	**96.9**	**84.0**	**98.2**
FG3D	$\mathcal{L}_{intra}$	0.23	0.15	0.65	20.2	24.4	10.4	34.7
	$\mathcal{L}_{intra} + \mathcal{L}_{inter}$	0.63	0.53	**0.84**	**83.1**	**91.6**	**73.0**	**95.5**

Table 5. *This table shows the maximum intra-class and minimum inter-class distances between object-identity classes after training, and also the test performance on single-view and multi-view object recognition and retrieval tasks for the three datasets, when training with and without the separation loss.*

the PAN encoder with and without the separation loss. We track how the intra-class and inter-class distances are optimized during training in Fig. 11, and also the performance on object recognition and retrieval tasks in Table 5.

In Fig. 11, we plot the maximum intra-class distance (d^{max}_{intra} in green), and the minimum inter-class distance between object-identity classes from the same category (d^{min}_{inter} in blue) during training to monitor the compactness and separability of object-identity classes respectively. These distances are computed using the object-identity embeddings and averaged over all objects. We also plot the ratio $\rho = \frac{d^{min}_{inter}}{d^{max}_{intra}}$ in red. A lower d^{max}_{intra}, and higher d^{min}_{inter} and ρ values would indicate embeddings of the same object-identity class are well clustered and separated from embeddings of other object-identity classes from the same category.

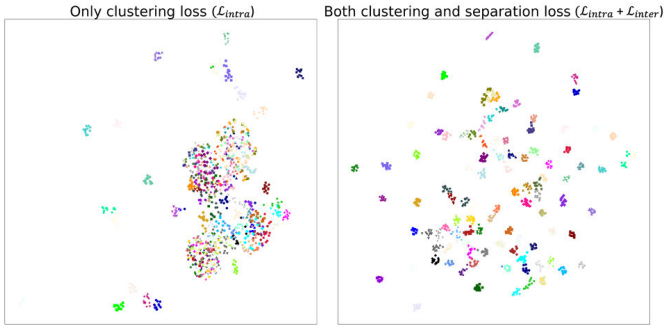


Figure 12. *We show the object-identity embeddings for a total of 100 objects from 5 categories of ModelNet-40 (20 objects from each category, such as tables, desks, chairs, stools, and sofas). The instances of each object-identity class is indicated by a unique color and shape. This figure illustrates that only clustering embeddings of the same object-identity classes is not sufficient to be able to distinguish between different object-identities, especially when there are many visually similar objects (left). Clustering the different views of the same object-identity classes, and simultaneously separating the different object-identity classes encourages learning more discriminative embeddings (right).*

We observe that training using the clustering loss ($\mathcal{L}_{intra}$) reduces the d^{max}_{intra} as it encourages clustering different views of the same object-identity together encouraging the network to learn pose-invariant features. However, only using the clustering loss also reduces d^{min}_{inter}, thereby reducing ρ as can be observed in the left of Fig. 11. Therefore, the object-identity classes are not well separated as can be seen in the left of Fig. 12, and this results in poor performance on object-level tasks.

Whereas, when training using the clustering and separation loss jointly ($\mathcal{L}_{intra} + \mathcal{L}_{inter}$), we observe in the right of Fig. 11 that ρ increases as the $d^{\text{min}}_{\text{inter}}$ decreases at a much slower rate than $d^{\text{max}}_{\text{intra}}$, and $d^{\text{min}}_{\text{inter}}$ eventually converges to a value beyond which it does not decrease substantially, indicating that our loss enforces separability between objects from the same category. For all the datasets, adding the separation loss yields significant performance improvement on object-level tasks, as can be seen in Table 5. This is because it enhances the inter-object separability that allows the encoder to learn more discriminative features to distinguish between visually similar objects. This can be observed in the right of Fig. 12, where each distinct object-identity class (indicated by a unique color and shape) can be more easily distinguished from other object-identity classes.

8. Optimizing Intra-class and Inter-class Distances in Single and Dual Spaces

As mentioned in the previous section, the pose-invariant object loss is designed to simultaneously enhance the inter-class separability and intra-class compactness of object-identity classes. In this section, we study how the distances between the samples in the object embedding space are optimized during training when learning single and dual embedding spaces.

For comparison, we show how these distances are optimized when learning representations in a single and dual embedding spaces, at the top and bottom of Fig. 13(A), (B),

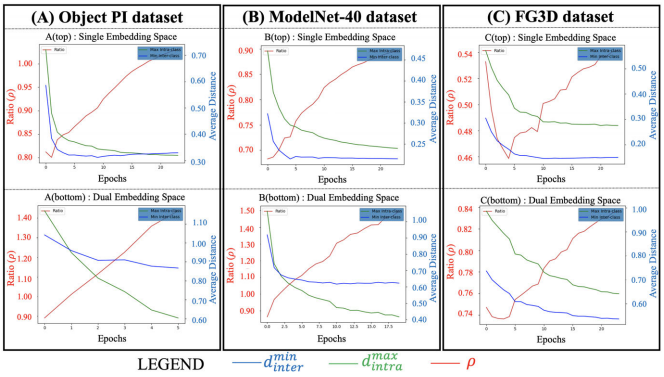


Figure 13. *For all three datasets, this figure illustrates that our pose-invariant object loss increases the ratio ρ that indicates better separability and compactness of object-identity classes. We observe that the values of the ratio ρ (in red) and the minimum inter-class distance d^{min}_{inter} (in blue) are higher when learning a dual space as compared to a single space. This indicates better separation between object-identity classes when learning in the dual space.*

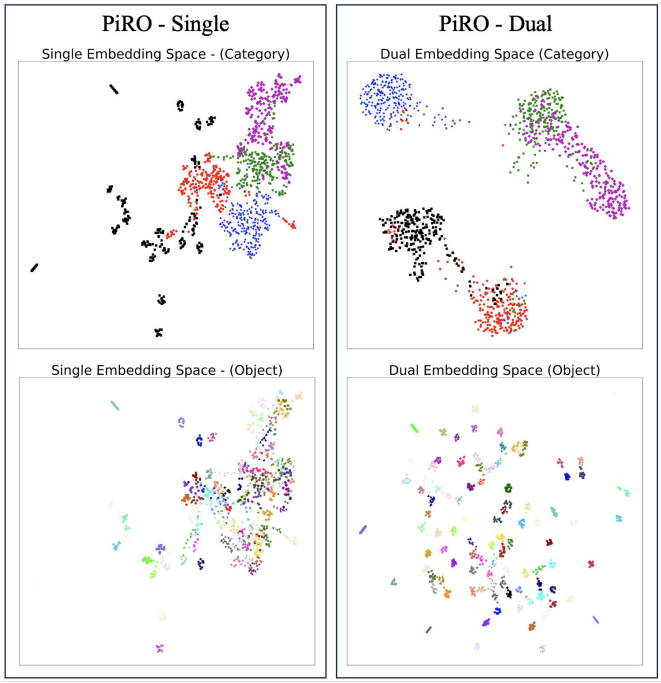


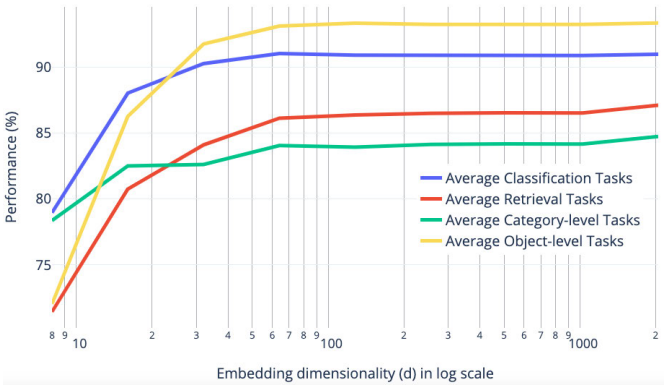
Figure 14. *UMAP visualization of the 100 objects from 5 different categories of ModelNet-40 for the single embedding space (left) and dual embedding space (right). The figure illustrates that decoupling the category and object-identity representations in separate spaces leads to better separability between categories in the category embedding space and object-identity classes in the object embedding space (right) as compared to learning representations in the same embedding space (left).*

and (C) for the ObjectPI, ModelNet-40, and FG3D datasets respectively. We observe that the values of ρ and the minimum inter-object distances ($d^{\text{min}}_{\text{inter}}$) are much higher in the dual space, which indicates that the object embeddings in the dual embedding space are better separated than those in the single embedding space, leading to better performance on object-based tasks, as shown in Table 6.

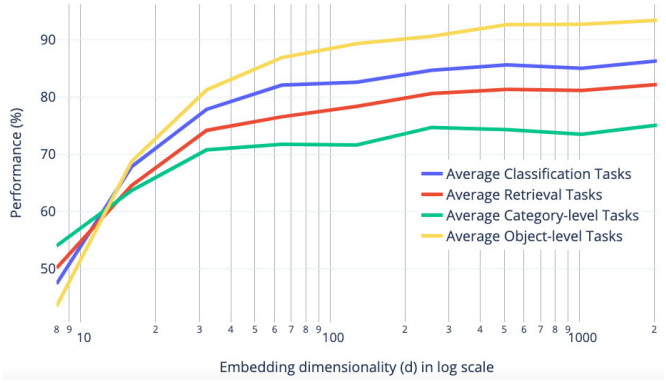
Fig. 14 illustrates this effect using UMAP visualizations of the embeddings in single and dual spaces. As mentioned in the paper, we jointly train our encoder using pose-invariant category and object-based losses. In the single embedding space, category-based losses aim to cluster embeddings of object-identity classes from the same category together, and in the same embedding space, the object-based loss aims to separate different object-identity classes from the same category. Due to these conflicting objectives in the same embedding space, object-identity classes are not separated well, as can be seen in Fig. 14 (left). In the dual space, the category and object representations are decoupled, and the category and object losses optimize the distances in the separate embedding spaces. As can be seen in Fig. 14 (right), the object and category embeddings are

Datasets	Embedding Space	Optimized distances during training			Test Performance on Object-level tasks			
					Classification (Acc. %)		Retrieval (mAP %)	
		$d^{max}_{intra}(\downarrow)$	$d^{min}_{inter}(\uparrow)$	$\rho(\uparrow)$	Single-view	Multi-view	Single-view	Multi-view
ObjectPI	Single	0.32	0.33	1.03	88.5	**98.0**	68.5	98.9
	Dual	0.60	**0.90**	**1.50**	**92.7**	**98.0**	**81.0**	**99.0**
ModelNet-40	Single	0.24	0.22	0.92	81.2	85.6	59.2	90.4
	Dual	0.41	**0.62**	**1.51**	**93.7**	**96.9**	**84.0**	**98.2**
FG3D	Single	0.29	0.16	0.55	26.2	31.0	15.7	42.9
	Dual	0.63	**0.53**	**0.84**	**83.1**	**91.6**	**73.0**	**95.5**

Table 6. *This table shows the maximum intra-class and minimum inter-class distances between object-identity classes after training when learning a single and dual embedding space. We observe that the d^{min}_{inter} and ρ values are higher in the dual embedding space indicating better separability of object-identity classes in the object embedding space. This yields better performance on object-level tasks for all the three datasets.*



(a) ModelNet-40 dataset



(b) ObjectPI dataset

Figure 15. *This graph illustrates the effect of embedding dimensionality on Average Classification and Retrieval performance for category and object-based tasks for ModelNet-40 (top) and ObjectPI (bottom) datasets.*

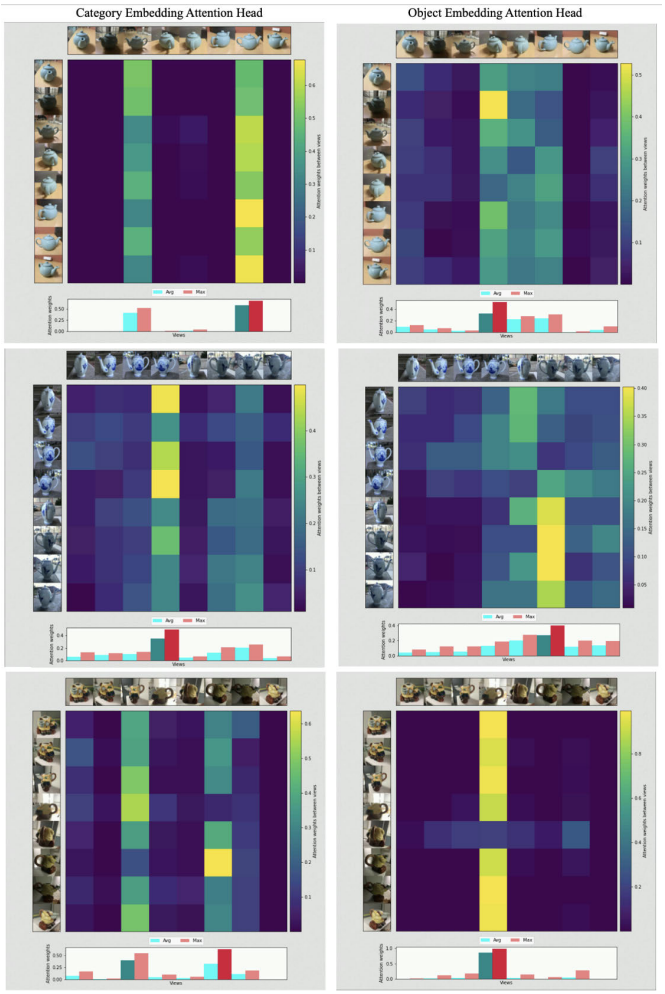


Figure 16. *Visualization of multi-view attention maps for the category and object self-attention layers.*

much better separated and we learn more discriminative embeddings overall in the dual space. This leads to significant performance improvements on object-based tasks, as can be seen in Table 6.

9. Embedding Dimensionality

For this experiment, we varied the embedding dimensionality from $d = 8, 16, \cdots, 2048$ for the category and object embedding space. We measured the performance of our method in terms of the average classification and retrieval performance as well as average performance on category-based and object-based tasks.

From Fig. 15, we observe that the performance on all four metrics improves with an increase in embedding dimensionality but beyond a certain embedding dimension, the performance only improves marginally. For the ModelNet-40 dataset, we observe that $d = 64$ for category-based tasks and $d = 128$ for object-based tasks is sufficient. For the ObjectPI dataset, we observe that $d = 256$ and $d = 512$ are sufficient for category-based and object-based tasks. We conjecture that higher embedding dimensionality is required for ObjectPI than ModelNet-40 as the embeddings for ObjectPI need to additionally capture color and texture information instead of just shape information for ModelNet-40. In general, the embedding dimensionality required for good performance on object-based tasks is higher than on category-based tasks as object embeddings need to capture more fine-grained details to differentiate between objects.

10. Multi-view Attention Maps

We plot the attention weights for the self-attention layers for the object and category embeddings in Figure 16. We observe that for category embeddings, the attention weights are higher for representative views that capture the overall shape of the kettle. All views of the object are correlated

to these representative views. For object embeddings, we observe that the attention weights are higher for the views that capture attributes related to the handle. This is possibly because the different kettles in the dataset have variations in the location (from the top or side) and shape of the handle.

11. Qualitative Retrieval Results

We show some qualitative object retrieval results on ObjectPI and ModelNet40 datasets in Figs. 17 and 18 respectively. The single-view object retrieval results show that given an arbitrary view of the object, our method can retrieve the other views of the same object correctly in Figs. 17 and 18. Despite variability in object appearance from different viewpoints, the presence of similar objects in the database as well as deformable objects (such as books, clothing, and so on), our method can retrieve objects with high precision.

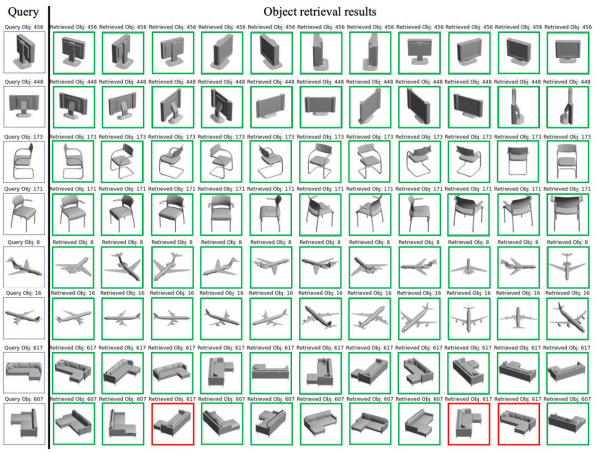


Figure 18. *For a single-view query in each row, the retrieved images of other views of the same object are shown on the right for the ModelNet40 dataset. The green and red bounding boxes indicate correct and wrong results respectively.*

Figure 17. *This figure shows our object retrieval results for the Object PI dataset. Given a single view query from an arbitrary pose on the left, the top-7 retrieved results are shown on the right in each row. Green bounding boxes indicate correct retrieval results and red boxes indicate incorrect results. In (a), we demonstrate that our framework can retrieve other views of the same object despite having similar objects in the test dataset. In (b), we demonstrate, that despite significant appearance changes under various pose transformations for different everyday objects, our framework can retrieve objects accurately.*

12. Pose-Invariant Category and Object-level Recognition and Retrieval Task Details

Since object appearance varies with viewpoints, the availability of multiple views of an object during inference helps in accurately recognizing objects. However, in most real-world applications, such as automatic checkout systems, robotic manipulation, content-based search, and product recognition and retrieval, only a single viewpoint of an object may be available at inference time. In an ideal scenario in which pose invariance has been achieved, the performance based on a single viewpoint should closely match that achieved when multiple viewpoints are available.

In our paper, we present recognition and retrieval performance results for both scenarios. That is, we consider both situations, one in which we only have a single view for an object, and two, when we have multiple views. The recognition performance is reported as classification accuracy and the retrieval performance is reported as mean average precision (mAP).

The details of category and object-level tasks are provided in the subsequent subsections.

12.1. Category-level tasks

Single-view or Multi-view category recognition: These tasks predict the category from a single view or a set of object views respectively. *Single-view or Multi-view category retrieval:* The goal of these tasks is to retrieve images from the same category as the query object from a single view or multiple views respectively. These are the same as [7].

12.2. Object-level tasks

In a practice that is common in solving problems in face and cross-view recognition, we evaluate pose-invariant recognition and retrieval by splitting the *test* dataset into two disjoint parts, 'probe' and 'gallery', with non-overlapping views for each object in both parts. For *Single-view object-level tasks*, the dataset is split such that for every object, each view is selected once as a probe image and the remaining views are selected as gallery images. The results averaged over all splits are reported. For *Multi-view object-level tasks*, the dataset is split equally into two halves, with one-half used as the gallery and the other half used as the probe. With the data being split in this manner, the two halves capture the object from opposite sides in the view-space.

The task details are as follows. *Single-view or Multi-view object recognition:* The goal is to recognize single-view or multi-view probe images by correctly matching them to the images of the same object-identity in the gallery dataset. *Single-view or Multi-view object retrieval:* Given a single view or a partial set of views of an object from the probe split as a query, this task aims to retrieve other views of the same object-identity (as the object in the query) from the gallery split.