MutualVPR: A Mutual Learning Framework for Resolving Supervision Inconsistencies via Adaptive Clustering

Qiwen Gu[1] **Xufei Wang**[3] **Junqiao Zhao**[1,2] * **Siyue Tao**[1] **Tiantian Feng**[4]
Ziqiao Wang[1] **Guang Chen**[1,5]
[1]School of Computer Science and Technology, Tongji University, Shanghai, China
[2]MOE Key Lab of Embedded System and Service Computing, Tongji University, Shanghai, China
[3]Shanghai Research Institute for Intelligent Autonomous System, Tongji University, Shanghai, China
[4]School of Surveying and Geo-Informatics, Tongji University, Shanghai, China
[5]Shanghai Innovation Institute
{2432178, tjwangxufei, zhaojunqiao, 2331923, Fengtiantian}@tongji.edu.cn
{ziqiaowang, guangchen}@tongji.edu.cn

Abstract

Visual Place Recognition (VPR) enables robust localization through image retrieval based on learned descriptors. However, drastic appearance variations of images at the same place caused by viewpoint changes can lead to inconsistent supervision signals, thereby degrading descriptor learning. Existing methods either rely on manually defined cropping rules or labeled data for view differentiation, but they suffer from two major limitations: (1) reliance on labels or handcrafted rules restricts generalization capability; (2) even within the same view direction, occlusions can introduce feature ambiguity. To address these issues, we propose MutualVPR, a mutual learning framework that integrates unsupervised view self-classification and descriptor learning. We first group images by geographic coordinates, then iteratively refine the clusters using K-means to dynamically assign place categories without orientation labels. Specifically, we adopt a DINOv2-based encoder to initialize the clustering. During training, the encoder and clustering co-evolve, progressively separating drastic appearance variations of the same place and enabling consistent supervision. Furthermore, we find that capturing fine-grained image differences at a place enhances robustness. Experiments demonstrate that MutualVPR achieves state-of-the-art (SOTA) performance across multiple datasets, validating the effectiveness of our framework in improving view direction generalization, occlusion robustness. The code can be found at `https://github.com/Gucci233/MutualVPR`.

1 Introduction

Visual Place Recognition (VPR) is the task of determining a previously visited location from a query image by matching it against a database of geo-tagged reference images. It serves as a key component in long-term localization and loop closure for autonomous systems such as mobile robots [10, 11, 32] and self-driving vehicles [12, 17].

Existing VPR methods either utilizes contrastive learning [4, 30, 20, 15, 2] or classification-based learning [26, 22, 6, 7] to learn the place representation. Contrastive learning-based approaches

[*] Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

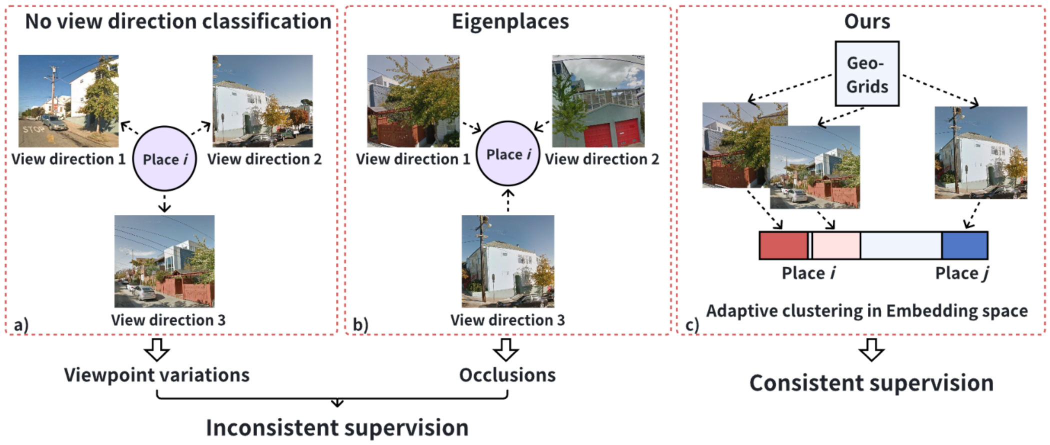

Figure 1: **The problem of inconsistent supervision in existing VPR researches.** The proposed mutual-learning framework define place labels by adaptive clustering in embedding space, enforcing supervision consistency.

facilitate the learning of robust and discriminative descriptors. However, they depend heavily on hard sample mining, which incurs significant computational cost and limits scalability to large-scale. Classification-based VPR methods divide the environment into spatial grids based on geographic coordinates, assigning each grid cell a unique class label, therefore, avoiding the need for expensive sample mining.

However, as shown in Figure 1 a), views captured from the same location but in different directions can result in drastically different visual scenes. If view direction is not considered, such views are treated as originating from the same place, which may introduce inconsistent supervision signals.

CosPlace [6] first attempts to address this problem by manually defining class labels based on view directions. However, this method does not provide explicit guarantees of intra-class visual similarity or inter-class visual distinctiveness. As a result, view variation may still persist. EigenPlaces [7] proposes a cropping strategy with the assumption that views toward a common reference point are similar. However, as shown in Figure 1 b), real-world scenes often involve occlusions from buildings, vehicles, and other structures. Such occlusions can lead to significant visual differences thus introduce inconsistent supervision. Consequently, visually dissimilar scenes may be incorrectly grouped under the same label. Such supervision inconsistency misaligns the supervisory signal with the true visual similarity, thereby undermining the model's ability to learn robust and discriminative features.

To address the problem of supervision inconsistency, we propose MutualVPR, a mutual learning framework that jointly refines image descriptors from the same geo-grids and view classification as shown in Figure 1 c). Unlike prior methods that rely on fixed or heuristically defined place labels, MutualVPR dynamically updates both feature representations and place label assignments through iterative, feature-driven mutual learning. This co-evolution enables the system to align semantic content with supervision more effectively.

Our main contributions can be summarized as follows:

- We propose a mutual learning framework where feature learning and clustering co-evolve, effectively mitigating supervision inconsistency.
- A adaptive clustering strategy dynamically refines pseudo-labels based on visual semantics, handling view directions and occlusions without orientation labels.
- Our method achieves state-of-the-art (SOTA) performance on challenging VPR benchmarks, demonstrating robustness to diverse appearance variations and inconsistent supervision.

2 Related Works

2.1 Contrastive Learning-based VPR

Recent advances in VPR have largely benefited from deep learning-based approaches. Methods such as NetVLAD [4] pioneered the use of trainable aggregation layers to produce compact and discriminative global image descriptors. Contrastive learning [8, 24] has become prevalent, employing triplet-based or contrastive objectives to encourage the model to learn discriminative representations.

However, images taken at the same location may be captured from very different view directions—resulting in large visual appearance gaps between positive samples. Conversely, images from nearby but distinct locations may look similar due to aligned view directions, increasing the risk of erroneous negatives. This mismatch between place labels and visual similarity can mislead contrastive objectives and degrade performance.

To mitigate the impact of view direction variation, MixVPR [2], CricaVPR [20], SALAD [15] and BoQ [3] train on the GSV-Cities dataset [1], where all images within the same class share a consistent view direction. By ensuring this, these methods reduce inconsistent supervision.

Sample mining strategies such as GCL [19] and Clique Mining (CM) [14] further enhance learning: GCL assigns graded similarity labels to reduce supervision noise, while CM forms batches of very similar images to create harder training samples.

Despite these improvements, all these approaches still face challenges when images from the same location exhibit diverse view directions or occlusions, limiting their generalization under extreme conditions.

Other methods [9, 13, 33, 21] aim to enhance robustness by incorporating local feature matching, but they often rely on a two-stage process, which incurs significant computational overhead.

2.2 Classification-based VPR

Despite their [2, 20, 15, 3, 18] success, contrastive learning-based methods rely on hard sample mining, which increases training complexity and computational overhead. An alternative approach is to formulate VPR as a classification problem, reducing the need for explicit pairwise comparisons. Methods such as DaC [28] directly partitions images into grids based on their geographic coordinates, training the feature extractor by classifying images into their corresponding grids. However, it does not take into account the variations in viewpoint within each grid. Furthermore, CosPlace [6] and EigenPlaces [7] categorize images into location and view-based classes, allowing efficient training with categorical cross-entropy loss.

CosPlace[6] partitions the dataset into geographic cells and classifies images based on view direction labels. However, this rule-based scheme overlooks visual similarity among samples within the same cell. As a result, it often assigns semantically similar images to different class and vice versa, leading to inconsistent supervision.

EigenPlaces [7] introduces a classification scheme based on the Singular Value Decomposition (SVD) of image locations, grouping images that share a common reference point. A key advantage of this approach is its independence from manually defined view direction labels. However, the underlying assumption—that images oriented toward the same reference point share similar visual content—often fails in urban environments due to occlusions caused by buildings, vehicles, or vegetation. Consequently, such scenes may exhibit substantial visual differences despite similar viewing intent, resulting in supervision inconsistencies that undermine classification reliability.

3 Problem Analysis

To better understand the issue of supervision inconsistency in classification-based VPR methods, we visualize the image descriptors extracted by CosPlace [6], EigenPlace [7], and our method using t-SNE.

As shown in Figure 2, each View corresponds to an image captured at a specific geographic position and camera orientation, while a Class denotes a group of views considered to represent the same place.

In practice, class labels can be assigned based on orientation labels (as in CosPlace), where each direction corresponds to a predefined class. However, this scheme often fails to align with true scene semantics, leading to inconsistent supervision when visually similar views fall into different classes. Alternatively, labels derived from descriptor clustering group images by visual similarity, naturally ensuring semantic consistency.

Such supervision inconsistencies mainly manifest as view variations and occlusions. View variation can be further divided into two types:

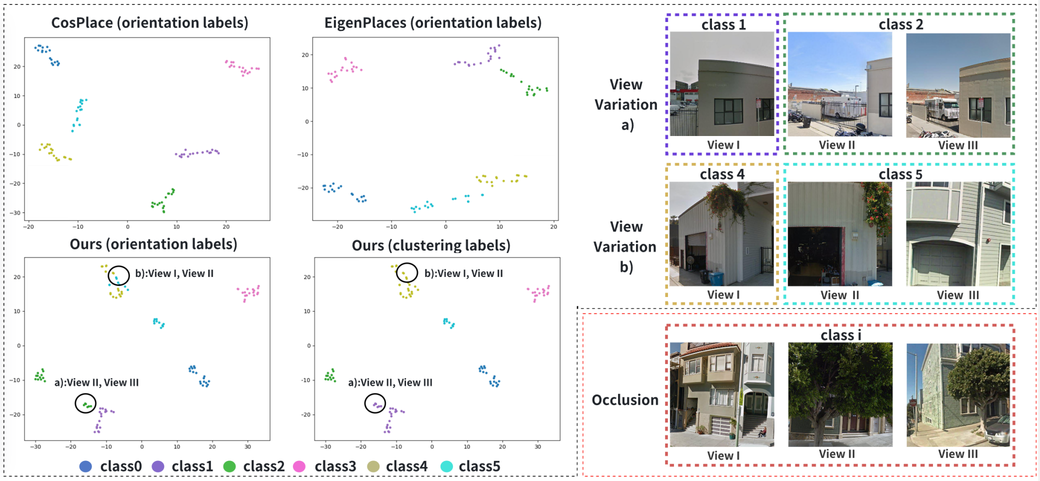

Figure 2: **Supervision Inconsistency in Classification-based Methods.** The left panel shows t-SNE visualizations of image descriptors extracted from a single geo-grid using different methods on the SF-XL dataset. “orientation labels” indicate samples colored according to their assigned view directions, while “clustering labels” refers to labels obtained by applying K-means clustering to the descriptors. The top-right panel illustrates issues related to view variation, using image examples drawn from samples in the t-SNE visualization on the left. The bottom-right panel illustrates occlusion induced issues, using image example from the same reference point.

View variation a): Views with large scene overlap are assigned to different labels due to orientation labels based on view direction. In panel a), Views II and III are labeled as Class 2, while the highly similar View I is labeled as Class 1. As shown in the t-SNE plots of CosPlace and EigenPlaces, their features directly inherit this flawed supervision, forming two separate clusters despite the strong visual overlap. When our learned features are colored by orientation labels, this artificial split remains; however, coloring the same features by clustering results yields a single, coherent group. This contrast highlights the core issue — the conflict between fixed directional supervision and actual visual semantics.

View variation b): The opposite inconsistency occurs when visually distinct views are assigned the same label. In panel b), Views II and III share Class 5 despite clear visual differences, while View I (Class 4) is semantically closer to View II. Feature visualizations colored by orientation labels reflect this mistaken grouping, whereas clustering-based coloring reorganizes them into more meaningful, semantically consistent clusters. This again exposes the mismatch between visual reality and rigid label assignment, underscoring the need for adaptive supervision.

Occlusion-induced inconsistencies are demonstrated by “Occlusion” in Figure 2, where images captured from different viewpoints but oriented toward the same reference point may exhibit significantly different visual content due to obstacles. In the "Occlusion" panel, Views I, II and III are captured from the same geo-grid but are occluded by trees or different buildings, leading to distinct visual content. Although these views share the same reference point, their semantic content diverges. This violates the assumptions of EigenPlace, resulting in supervision inconsistency.

We therefore argue that effective supervision for VPR should reflect semantic similarity rather than rely solely on spatial proximity or fixed directional assumptions. Without semantically consistent supervision, models are prone to learning unstable features, reducing their ability to generalize across complex, realworld scenarios.

4 Approach

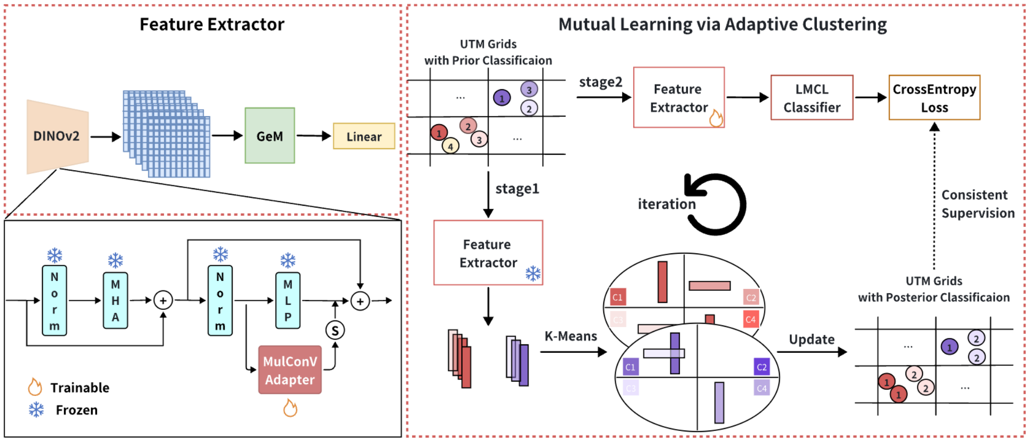

Figure 3: **Mutual Learning Framework via Adaptive Clustering.** We initialize spatial grids using UTM coordinates and assign coarse intra-grid categories. Features are extracted using DINOv2[23] with adapter and GeM [25] pooling, while adaptive clustering where iterative K-means is guided by LMCL loss dynamically refines view direction categories within grids. Clusters and features co-evolve: updated clusters supervise feature learning (stage 1), and improved features guide reclustering (stage 2), enabling robust supervision under occlusions and view direction changes.

The proposed MutualVPR framework integrates unsupervised view self-classification with joint descriptor training in a mutual learning paradigm (as shown in Figure 3). The key lies in its adaptive clustering mechanism, which iteratively refines place categories throughout training.

4.1 Feature Encoder

MutualVPR is built upon DINOv2 [23], and incorporates the MulConv adapter [20] to enhance the model's capacity for robust feature representation. MulConv adopts a bottleneck structure with three parallel convolutional branches operating at different receptive fields, enabling the extraction of multi-scale features. This design improves the model's ability to handle variations in object scale and spatial context, which is beneficial in environments with diverse structural patterns. It can be expressed as:

$$\begin{aligned} z_l' &= \mathrm{MHA}\left(\mathrm{LN}\left(z_{l-1}\right)\right) + z_{l-1}, \\ z_l &= \mathrm{MLP}\left(\mathrm{LN}\left(z_l'\right)\right) + s \cdot \mathrm{Adapter}\left(\mathrm{LN}\left(z_l'\right)\right) + z_l'. \end{aligned} \tag{1}$$

where z_{l-1} denotes the input features from the previous layer, $\mathrm{LN}(\cdot)$ is Layer Normalization, and $\mathrm{MHA}(\cdot)$ refers to Multi-Head Attention. $\mathrm{MLP}(\cdot)$ stands for the feed-forward network, while $\mathrm{Adapter}(\cdot)$ represents the MulConv adapter module. The parameter s is a learnable scaling factor that controls the contribution of the adapter branch.

4.2 Mutual Learning via Adaptive Clustering

4.2.1 Place Label Initialization

Similar to CosPlace[6], but without using view direction labels, we first partition images into coarse location-based classes using UTM coordinates. At this stage, view direction variations within the same location are not considered. Formally, a coarse location class is defined as:

$$x = \left\{ \left\lfloor \frac{east}{M} \right\rfloor = e_i, \left\lfloor \frac{north}{M} \right\rfloor = n_j \right\}, \tag{2}$$

where M is a hyperparameter controlling the size of the grid. This step serves as an initialization of place labels, which are later refined through adaptive clustering.

4.2.2 Mutual Learning of Feature Extraction and Clustering

While UTM-based grouping ensures spatial locality, it fails to capture appearance variations caused by view direction changes and occlusions. To address this, we introduce a mutual learning framework that combines feature-aware supervision with adaptive clustering.

Within each coarse UTM region, we perform iterative K-means clustering based on learned descriptors, enabling view direction categories to adapt dynamically as feature representations evolve. Formally, view direction categories are defined as:

$$C = \{e_i, n_j, h \mid e_i, n_j \in x, h \in K\}, \tag{3}$$

where K controls the granularity of view direction partitioning. Unlike static classification methods with fixed view direction labels, our approach allows continuous refinement of view direction clusters, better aligning them with actual visual similarity.

As training progresses, the feature encoder and clustering process update each other iteratively. This self-correcting mechanism avoids error accumulation from early misassignments and ensures images are grouped into more consistent view direction categories over time.

Following CosPlace [6], we adopt Large Margin Cosine Loss (LMCL) [29] as our classifier to enforce discriminative feature learning. At inference time, descriptors are directly extracted from the feature encoder and used for image retrieval based on their distances.

4.3 Implementation Details

In the implementation, DINOv2's ViT-B 14 is employed as the backbone network, with all parameters frozen except for the Adapter. Input image size is resized to 504×504 to meet the input requirements of ViT-B 14. GeM pooling and a fully connected layer reduce the dimensionality to 512 for the final descriptor. For dataset classification, margin M is set to 10, and K clusters are set to 3. The feature extractor is initialized with a learning rate of 1e-5, while the classifier uses 1e-2. We employ the Adam optimizer and apply a cosine annealing scheduler to the feature extractor. Training is conducted for 50 epochs, each consisting of 10,000 iterations. To reduce training time due to the huge dataset, we divide all UTM classes from the coarse classification into eight groups, with each group serving as the training set for one epoch. In each epoch, one-fifth of the classes within the selected group is randomly chosen for feature-aware clustering.

During training, we found that increasing overlapping ratio of cropped images can enhance semantic continuity. For the multi-angle cropping strategy, we crop panoramic images from different starting angles every 60° to generate training data. In our experiments, the starting angles are set to 0° and 30°. Since all images are cropped from panoramas, adjacent images naturally share semantic content. In our work, adjacent classes—such as class 2 and 3—represent neighboring view directions.

5 Experimental Results

5.1 Research Questions

In this work, we focus on the supervision inconsistency problem in VPR. We aim to investigate the following research questions:

Q1: How does our method perform compared to SOTA VPR approaches across standard benchmarks?

Q2: How well does our method generalize to challenging dataset with occlusion?

Q3: How does our adaptive clustering compare to orientation labels in handling supervision inconsistency?

5.2 Datasets and Evaluation Metrics

For contrastive learning-based VPR methods, we use the GSV-Cities dataset [1] for training. For classification-based VPR methods, we use the SF-XL dataset [6] for training. The selected training set is a subset of about 0.9M panoramic images, following CosPlace. A multi-angle cropping strategy as describe in Section 4.3 is applied. Eigenplaces use all panoramic images of about 3.4M for cropping.

For evaluation, we adopt several widely used VPR benchmarks: Pitts30k-test, Pitts250k-test [4], MSLS-val [31], Tokyo 24/7 [27], and SF-XL-test v1 [6]. To assess the generalization capability of our method, we also evaluate it on SF-XL-occlusion [5] dataset. This dataset, originally designed to assess the robustness of VPR methods under severe occlusion, also exposes the limitations of conventional supervision strategies. The detailed statistics of the datasets can be seen in Appendix A.

We employ the standard Recall@K metric, defined as the ratio of correctly located queries to the total number of queries. Correct localization involves searching for positive examples by matching images within 25-meter radius threshold based on geographical coordinates.

The experiment is executed on a server with three NVIDIA RTX 3090 GPUs, using PyTorch for training and testing.

5.3 Comparison with Other Methods

Method	Desc.dim.	Train set	MSLS-val		Pitts30k		Pitts250k		Tokyo24/7		SF-XL-testv1	
			R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
NetVLAD	131072	GSV-Cities	82.8	90.3	87.0	94.3	89.1	94.8	69.5	82.5	-	-
GeM	2048	GSV-Cities	72.5	82.7	84.5	92.8	85.1	93.4	60.3	73.7	25.6	35.5
AnyLoc(ViT-B+GeM)	768	-	32.6	41.6	77.7	88.9	79.3	89.5	71.7	87.6	33.3	45.2
ConvAP	2048	GSV-Cities	81.5	87.5	89.7	95.2	91.2	96.4	74.6	83.2	41.1	53.0
MixVPR	4096	GSV-Cities	87.1	91.4	91.6	95.5	**94.3**	98.1	87.0	93.3	69.2	77.4
CricaVPR*	4096	GSV-Cities	90.0	95.4	94.9	97.3	–	–	93.0	97.5	-	-
CricaVPR$_1$	4096	GSV-Cities	88.5	95.1	91.6	95.7	**94.3**	**98.6**	89.5	94.6	72.8	80.1
CricaVPR$_1$ (PCA)	512	GSV-Cities	87.1	92.6	90.4	94.9	92.5	97.1	87.4	92.9	68.4	77.1
BoQ†	512	GSV-Cities	88.4	93.9	**93.1**	96.1	93.8	97.5	91.9	95.5	79.6	85.9
SALAD†	512+32	GSV-Cities	88.5	94.2	90.6	95.1	92.1	97.0	92.3	95.1	70.2	77.7
SALAD+CM†	512+32	MSLS+GSV-Cities	**90.4**	**96.2**	90.9	95.9	93.2	97.8	**92.8**	96.2	78.4	85.4
EigenPlaces	512	SF-XL	88.1	92.9	92.3	96.1	93.5	97.5	84.8	94.0	**83.8**	**89.6**
CosPlace	512	SF-XL	84.4	90.2	89.6	94.9	90.4	96.6	76.5	89.2	64.8	73.1
MutualVPR (Ours)	512	SF-XL	89.2	95.1	90.9	**96.4**	92.6	97.9	92.4	**96.6**	80.8	86.4

Table 1: **Comparison of various methods on multiple benchmark datasets.** The upper block lists contrastive learning methods, while the lower block lists classification-based methods. CricaVPR* denotes the results from the original paper relying on batch interaction, and CricaVPR$_1$ is our single-query variant. † indicates a retrained 512-D or 512+32-D version. Best results are shown in **bold**, second best are underlined.

To answer the research question Q1, we compare our method with SOTA VPR methods, including both contrastive learning-based and classification-based approaches. The former includes NetVLAD [4], GeM [25], AnyLoc [18], ConvAP [1],MixVPR [2], CricaVPR [20], BoQ [3], SALAD [15] and CM [14]. The latter includes EigenPlaces [7], and CosPlace [6].

It should be noted that our NetVLAD is trained on a ResNet-50 backbone, unlike the original version trained on VGG16. Nevertheless, its descriptor dimensionality is still very high, which prevents evaluation on the SF-XL-test set due to memory constraints when processing 2M samples. The comparison results are shown in Table 1. It shows that MutualVPR consistently delivers competitive or superior performance across multiple benchmarks, showing strong generalization capability.

For classification-based baselines, MutualVPR generally outperforms CosPlace across datasets, even when CosPlace uses ground-truth labels. This demonstrates the benefit of our adaptive clustering, which mitigates the limitations of orientation labels with fixed splits that may misrepresent visual similarity under viewpoint changes or occlusions. EigenPlaces, on the other hand, is specifically designed to handle extreme viewpoint variations. While it performs well on SF-XL-testv1, its performance drops on more diverse datasets such as Tokyo 24/7 and MSLS-Val. This is because EigenPlaces' strong focus on viewpoint invariance can limit its generalization to scenarios involving occlusions or other challenging conditions. In contrast, MutualVPR achieves a balance: it effectively handles viewpoint variations while remaining robust to occlusions and other extreme conditions, leading to better overall generalization. Supporting experiments validate this claim in Appendix D.

Among contrastive learning-based methods, those trained on the GSV-Cities dataset generally exhibit strong performance. This is largely attributed to the nature of GSV-Cities, which contains multiple captures of the same location under highly consistent viewpoints. Such data inherently mitigates supervision inconsistencies caused by view variations, allowing contrastive methods to learn more stable representations. However, achieving this level of performance requires large-scale, carefully curated data and extensive sample mining. Recent state-of-the-art methods, such as SALAD+CM, further leverage large and diverse training sets (MSLS + GSV-Cities) to achieve the highest average performance across multiple benchmarks.

In contrast, our MutualVPR is trained solely on SF-XL yet achieves robust and well-balanced performance across diverse conditions. Although not always the top performer on every individual benchmark, it ranks closely behind SALAD+CM in overall accuracy and even surpasses it on Pitts30k and Tokyo24/7 in terms of R@5. Moreover, unlike methods such as CricaVPR*, which rely on batch-level feature interaction to enhance localization accuracy and suffer a substantial drop in single-query inference (as indicated by CricaVPR$_1$), our approach maintains stable performance without requiring inter-sample dependencies, further underscoring its practical robustness.

5.4 Evaluate on Occlusion Dataset

To answer research questions Q2, we evaluate the generalization capability of our method using a dataset characterized by significant occlusions, which further reflects the impact of inconsistent supervision on retrieval performance. Table 2 presents the retrieval performance on the SF-XL-Occlusion dataset, where each query is affected by occlusion.

Our method achieves 47.4% R@1, significantly outperforming all baselines, including EigenPlaces (36.8%) and CosPlace (32.9%), as well as several contrastive learning methods such as MixVPR (30.3%) and SALAD (31.6%). Among the contrastive learning baselines, CricaVPR$_1$ (40.8%), SALAD+CM (40.8%), and BoQ (38.2%) show competitive performance, but still remain below our method for top-k retrieval metrics. These results demonstrate that our approach maintains robust retrieval under heavy occlusion.

Method	**Desc.dim.**	**SF-XL-Occlusion**			
		R@1	**R@5**	**R@10**	**R@20**
GeM	2048	11.8	15.8	17.1	22.4
AnyLoc(ViT-B+GeM)	768	6.6	14.5	19.7	26.3
ConvAP	2048	23.7	26.3	28.9	31.6
MixVPR	4096	30.3	35.5	38.2	44.7
CricaVPR$_1$	4096	40.8	51.3	54.6	59.9
BoQ†	512	38.2	50.0	53.3	59.2
SALAD†	512+32	31.6	42.1	46.1	51.3
SALAD+CM†	512+32	40.8	53.7	58.3	61.3
EigenPlaces	512	36.8	51.8	56.6	59.2
CosPlace	512	32.9	43.4	46.1	48.7
No Classification	512	17.1	25.0	26.3	31.6
MutualVPR (Ours)	512	**47.4**	**65.8**	**71.1**	**73.7**

Table 2: **Comparison on SF-XL-Occlusion.** Each query in SF-XL-Occlusion is affected by occlusion, making it suitable for testing robustness under missing visual cues. The upper block lists contrastive learning methods, and the lower block lists classification-based ones. “No Classification” indicates that no intra-grid classification is applied—images within the same grid are treated as belonging to the same class. Best results are shown in **bold**, and second best are underlined.

Our strong performance under occlusion stems from two key advantages:

Adaptive Supervision Consistency: Unlike static supervision methods (e.g. CosPlace, EigenPlaces), our approach refines class assignments through iterative, feature-aware clustering. This process corrects initial supervision errors caused by occlusions or view changes, progressively aligning features from occluded and unoccluded samples belonging to the same place. As a result, the model learns to associate semantically similar scenes across varying view directions, improving label consistency and robustness.

See Appendix B.1 for a visual example showing a misclassified occluded query being reassigned to the correct class after training.

Semantic Proximity within Class: Our method brings semantically similar views closer in feature space, even when they originate from different view directions. This contrasts with rigid label-based schemes, where visually similar scenes are separated due to orientation labels.

This semantic proximity benefits retrieval: even if an occluded query is not correctly classified, it can still be retrieved as long as its feature lies within the threshold distance.

We conducted experiments showing that our method achieves consistently lower inter-class distances than CosPlace and EigenPlaces across adjacent view directions. Quantitative analysis and distance comparisons are presented in Appendix B.2.

These findings suggest that by iteratively updating class assignments based on feature similarity, our approach naturally brings semantically similar scenes—despite view differences or occlusions—closer in the feature space. This results in more compact and coherent class boundaries.

In contrast, these methods which rely on rigid, manually defined labels often split visually similar scenes into separate classes, creating artificial gaps in the feature space. Our method mitigates such fragmentation, enabling smoother transitions between adjacent classes and enhancing retrieval robustness.

5.5 Ablations

5.5.1 Adaptive vs Static Label Supervision

To answer research question Q3, we compare manually assigned view direction labels with our adaptive clustering approach across various settings.

- Fixed view direction Labels (CosPlace) : Using predefined view direction labels from geographic metadata.
- Fixed view direction Labels (CosPlace) + Cropping: Applying multi-angle cropping while still relying on predefined view direction labels.
- Adaptive Clustering (Ours): Using our adaptive clustering but training on raw images without multi-angle cropping.
- Adaptive Clustering (Ours) + Cropping: Incorporating both self-adaptive clustering and multi-angle cropping.

Method / Diff.	**Backbone**	**Cropping strategy**	**Tokyo24/7**		**SF-XL-testv1**	
			R@1	**R@5**	**R@1**	**R@5**
CosPlace	ResNet50	0°	76.5	89.2	64.8	73.1
CosPlace	ResNet50	30°	80.1	90.2	70.1	80.6
CosPlace	ResNet50	0°+30°	85.1	92.4	81.1	86.2
MutualVPR(Ours)	ResNet50	0°	82.9	90.2	74.5	83.1
MutualVPR(Ours)	ResNet50	30°	81.6	91.0	72.4	81.6
MutualVPR(Ours)	ResNet50	0°+30°	85.4	92.5	74.8	82.5
CosPlace	DINOv2	0°	90.2	95.3	76.6	86.3
CosPlace	DINOv2	30°	89.8	95.0	75.9	86.1
CosPlace	DINOv2	0°+30°	91.0	95.8	79.1	86.3
MutualVPR(Ours)	DINOv2	0°	91.1	97.1	77.0	84.6
MutualVPR(Ours)	DINOv2	30°	89.9	96.0	78.1	84.4
MutualVPR(Ours)	DINOv2	0°+30°	92.1	96.5	80.8	86.4

Table 3: **Performance Comparison of Ground Truth and Cropping.** CosPlace represents a method that uses ground-truth and our method represents mutual learning frame. 0° and 30° indicate the starting angles when cropping the panorama.

We conducted experiments using both DINOv2 and ResNet50 as backbones. For each starting angle, we cropped the panoramic image into six evenly spaced views; e.g. with starting angles of 0° and 30°, this results in a total of 12 cropped images. In our method using ResNet50, we clustered all views into six classes ($K = 6$), while for DINOv2, we followed the same procedure but set $K = 3$.

While CosPlace sees clear gains from multi-angle cropping on SF-XL-testv1 (R@1 up to 81.1%), its performance on Tokyo24/7 (R@1 at 85.1%) reveals limited generalization, likely due to overfitting.

In contrast, MutualVPR achieves consistently strong results across both datasets. Both methods benefit from the multi-angle cropping strategy, which increases semantic continuity, but MutualVPR gains more thanks to its adaptive clustering that better aligns semantically similar views.

5.5.2 Cluster Number

The number of clusters, K, is a hyperparameter that determines the granularity of dataset partitioning using K-means. With fewer clusters, the images within each class tend to be more compact and exhibit higher similarity in the feature space. Conversely, a larger K results in finer partitioning, increasing the diversity of images within the feature space. In our experiments, we evaluated $K = 1, 3, 6$, and the corresponding results are shown in Table 4.

Backbone	K	Tokyo24/7		SF-XL-testv1	
		R@1	R@5	R@1	R@5
ResNet50	1	68.3	84.8	52.1	62.0
ResNet50	3	81.0	89.8	73.9	81.4
ResNet50	6	82.9	90.2	74.5	83.3
DINOv2	1	80.6	91.4	61.1	71.1
DINOv2	3	91.1	97.1	77.0	84.6
DINOv2	6	86.0	94.0	70.9	79.6

Table 4: **Different Cluster Numbers.** Trained on the original dataset, results with a descriptor dimension of 512. $K = 1$ can be considered as not classifying the dataset, while $K = 6$ aims to match the number of cluster labels with the ground truth clustering labels.

As expected, the case without clustering ($K = 1$) yields the worst performance, demonstrating the effectiveness of incorporating view direction classification. For the ResNet50 backbone, performance is highest when $K = 6$, though the improvement over $K = 3$ is marginal. In contrast, for the DINOv2 backbone, optimal performance is achieved with $K = 3$, outperforming $K = 6$ by approximately 10%. These results suggest that the optimal number of clusters depends not only on the dataset but also on the backbone. Determining an appropriate K for a given dataset and model remains an open question in the context of the proposed method.

More ablation studies can be found in the Appendix C.

6 Conclusion and Future Work

We proposed MutualVPR, a mutual learning framework that addresses supervision inconsistencies in VPR caused by view direction variations and occlusions. By dynamically refining view direction categories through adaptive clustering guided by feature learning, our method eliminates reliance on orientation labels and achieves semantically consistent supervision. Extensive experiments show that MutualVPR achieves robust and generalizable performance across diverse and challenging datasets, validating the effectiveness of our adaptive clustering strategy for real-world VPR applications.

A limitation of our approach is the fixed cluster number K, which may not fully capture varying view direction distributions. Since classification and descriptor learning are mutually reinforced, a static K may limit model's adaptability. Our studies underscore its impact on performance, suggesting the need for dynamic adjustment. Future work will explore adaptive clustering to optimize K based on dataset characteristics, enhancing the synergy between classification and representation learning.

Acknowledgments and Disclosure of Funding

This work is supported by the National Key Research and Development Program of China (No. 2020YFA0711402).

References

[1] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguere. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022.

[2] Amar Ali-Bey, Brahim Chaib-Draa, and Philippe Giguere. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2998–3007, 2023.

[3] Amar Ali-Bey, Brahim Chaib-draa, and Philippe Giguère. Boq: A place is worth a bag of learnable queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17794–17803, 2024.

[4] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[5] Giovanni Barbarani, Mohamad Mostafa, Hajali Bayramov, Gabriele Trivigno, Gabriele Berton, Carlo Masone, and Barbara Caputo. Are local features all you need for cross-domain visual place recognition? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6155–6165, 2023.

[6] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4878–4888, 2022.

[7] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. Eigenplaces: Training viewpoint robust models for visual place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11080–11090, 2023.

[8] Gabriele Moreno Berton, Valerio Paolicelli, Carlo Masone, and Barbara Caputo. Adaptive-attentive geolocalization from few queries: A hybrid approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2918–2927, 2021.

[9] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 726–743. Springer, 2020.

[10] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3223–3230. IEEE, 2017.

[11] Zetao Chen, Lingqiao Liu, Inkyu Sa, Zongyuan Ge, and Margarita Chli. Learning context flexible attention model for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 3(4):4015–4022, 2018.

[12] Anh-Dzung Doan, Yasir Latif, Tat-Jun Chin, Yu Liu, Thanh-Toan Do, and Ian Reid. Scalable place recognition under appearance change for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9319–9328, 2019.

[13] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14141–14152, 2021.

[14] Sergio Izquierdo and Javier Civera. Close, but not there: Boosting geographic distance sensitivity in visual place recognition. In *European Conference on Computer Vision*, pages 240–257. Springer, 2024.

[15] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 17658–17668, 2024.

[16] Tong Jin, Feng Lu, Shuyu Hu, Chun Yuan, and Yunpeng Liu. Edtformer: An efficient decoder transformer for visual place recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[17] Shubham Juneja, Povilas Daniušis, and Virginijus Marcinkevičius. Visual place recognition pre-training for end-to-end trained autonomous driving agent. *IEEE access*, 11:128421–128428, 2023.

[18] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1286–1293, 2023.

[19] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Data-efficient large scale place recognition with graded similarity supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23487–23496, 2023.

[20] Feng Lu, Xiangyuan Lan, Lijun Zhang, Dongmei Jiang, Yaowei Wang, and Chun Yuan. Cricavpr: Cross-image correlation-aware representation learning for visual place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16772–16782, 2024.

[21] Feng Lu, Lijun Zhang, Xiangyuan Lan, Shuting Dong, Yaowei Wang, and Chun Yuan. Towards seamless adaptation of pre-trained models for visual place recognition. *arXiv preprint arXiv:2402.14505*, 2024.

[22] Eric Muller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–579, 2018.

[23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[24] Guohao Peng, Yufeng Yue, Jun Zhang, Zhenyu Wu, Xiaoyu Tang, and Danwei Wang. Semantic reinforced attention learning for visual place recognition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13415–13422. IEEE, 2021.

[25] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1655–1668, 2018.

[26] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 536–551, 2018.

[27] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1808–1817, 2015.

[28] Gabriele Trivigno, Gabriele Berton, Juan Aragon, Barbara Caputo, and Carlo Masone. Divide&classify: Fine-grained classification for city-wide visual geo-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11142–11152, 2023.

[29] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[30] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. Transvpr: Transformer-based place recognition with multi-level attention aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13648–13657, 2022.

[31] Frederik Warburg, Søren Hauberg, Manuel López-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[32] Ming Xu, Niko Snderhauf, and Michael Milford. Probabilistic visual place recognition for hierarchical localization. *IEEE Robotics and Automation Letters*, 6(2):311–318, 2020.

[33] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng Wang. R2former: Unified retrieval and reranking transformer for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19370–19380, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations and future work are discussed in the Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setting and details are described in the Section 5 and some experimental details is shown in Appendix B and Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open Access to Data and Code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code will be submitted along with the Supplementary Material. And code also can be found at `https://github.com/Gucci233/MutualVPR`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting and details are described in the Section 4.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports the mean and standard deviation of the results in the tables and figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides the details of the compute resources in the last paragraph of the Section 5.2 and analyzed the parameter quantities of different backbones in the Appendix C.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and ensured that our research conforms to it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release data or models that have a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for Existing Assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits the original owners of the assets and mentions the license and terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We publish our source code, and the new assets are well documented in this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Dataset Details

We train on several large-scale datasets, including SF-XL[6] for classification-based methods , GSV-Cities [1] for contrastive learning, and evaluate on Pitts30k-test [4], MSLS-val [31], Tokyo 24/7 [27].

GSV-Cities is a large-scale dataset containing 560k images depicting 67k unique places captured from consistent viewpoints, each labeled with geographic coordinates. SF-XL consists of images cropped from panoramic views at the same location, covering diverse viewing angles and acquisition years, making it suitable for learning viewpoint- and time-robust representations.

A summary of above datasets is provided in Table 5.

Dataset	Database	Query
SF-XL-train	5.6M	
SF-XL-test	2M	1000
SF-XL-val	8K	8064
SF-XL-occlusion	2M	76
GSV-Cities-train	560K	
Pitts30k-test	10K	6818
MSLS-val	18.9k	740
Tokyo24/7-test	76K	315

Table 5: **Experimental dataset statistics.**

B Supervision Correction and Feature Distance Analysis

To further support our claims on supervision correction and semantic proximity, we provide additional visualizations and feature-level analysis of our method.

B.1 Visualization of Supervision Correction

To demonstrate how adaptive clustering resolves initial supervision inconsistencies (e.g., caused by occlusion), we visualize clustering results before and after training.

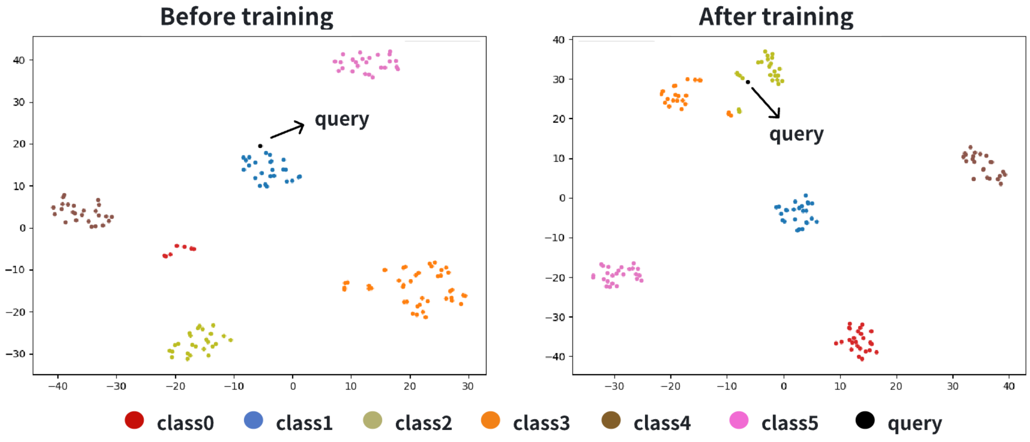

Figure 4: **The t-SNE visualization of clustering results on SF-XL-Occlusion before and after training.** The query (black dot) is reassigned from class 1 to class 2, correcting its initial misclassification.

As shown in Figure 4, we select a query (black dot) and its neighboring samples from the same geo-grid cell. Initially, the query is assigned to class 1, but after training, it transitions to class 2. These two classes correspond to adjacent view directions with overlapping semantic content.

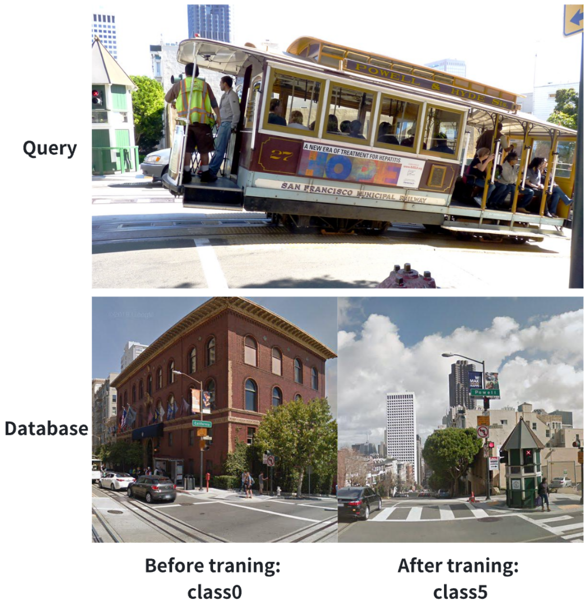

Figure 5: **Close-up of the query and its nearest samples.** Visual inspection shows class 2 has stronger semantic similarity to the query.

Figure 5 shows a visual comparison of samples in class 1 and class 2, where class 2 clearly exhibits stronger visual similarity with the query. This supports our claim that adaptive clustering can realign mislabeled samples by leveraging feature similarity during training, improving robustness to occlusion and supervision noise.

This demonstrates that our adaptive clustering effectively mitigates the impact of occlusions during training by refining feature-grouping over time. It enables the model to build robust associations between partially occluded and unobstructed views from the same location.

B.2 Feature Distance Analysis Across Classes

We further analyze how our adaptive clustering improves semantic continuity between classes by comparing feature distances of samples from adjacent categories.

To ensure fairness, we use SOTA method EigenPlaces as a proxy for selecting image pairs with adjacent labels and minimum mutual distances. As shown in Figure 6, we compare the pairwise feature distances obtained by our method and CosPlace.

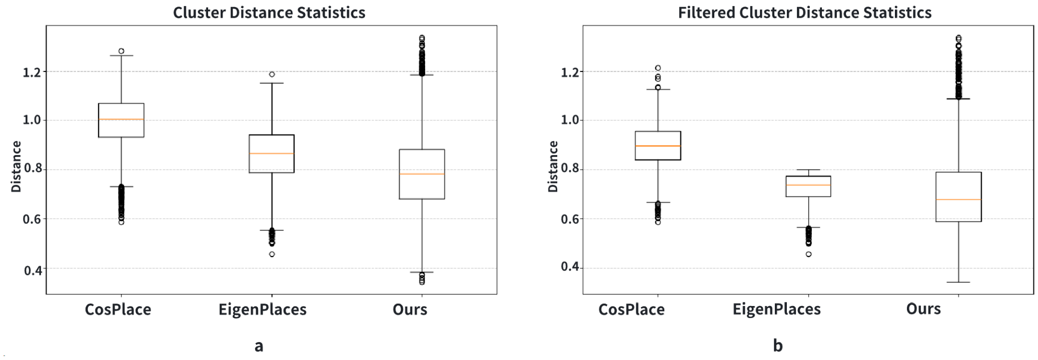

Figure 6: **Visualization results.** Feature distance comparisons between adjacent classes, using EigenPlaces as a proxy. Our method shows tighter clustering both overall (a) and for similar pairs with $d < 0.8$ (b), indicating better feature continuity across classes.

The results show that our method consistently achieves the smallest feature distances across adjacent view directions Even when we focus on sample pairs with closer feature distances (< 0.8), as shown in Figure 6 b. Our method still maintains closer feature relationships.

Our method consistently yields lower feature distances than CosPlace and EigenPlaces, indicating that our approach encourages more compact and semantically smooth transitions between neighboring classes. This explains why even occluded queries, if misclassified, can still retrieve the correct place as long as the distance remains within the retrieval threshold.

C More Studies

C.1 Comparison of High-Dimensional Descriptor

Method	**Desc.dim.**	**SF-XL-Occlusion**				**SF-XL-testv1**			
		R@1	**R@5**	**R@10**	**R@20**	**R@1**	**R@5**	**R@10**	**R@20**
BoQ	8192	49.8	65.8	70.3	**75.9**	82.3	87.9	**91.3**	**92.9**
SALAD	8192+256	45.4	62.2	67.1	74.6	82.4	86.6	89.2	90.0
SALAD+CM	8192+256	46.1	57.9	64.5	68.4	80.6	85.3	87.7	89.1
MutualVPR (Ours)	8192	**51.4**	**67.5**	**72.8**	75.8	**82.8**	**88.9**	90.8	91.4

Table 6: **Comparison on SF-XL-Occlusion and SF-XL-testv1.** Comparison with high-dimensional DINOv2-based methods. Our method shows limited improvement when increasing dimensions, but still achieves competitive performance. Best results are shown in **bold**, and second best are underlined.

Since our method already achieves excellent performance with a relatively low descriptor dimension (512), we further conduct a fair comparison with SOTA methods under similar dimensional settings, as shown in Table 6. The original BoQ model has a very high dimensionality of 12288, making it infeasible to evaluate on the SF-XL dataset. Therefore, we reduce its projection dimension from 384 to 256, resulting in a total descriptor dimension of 8192.

To ensure fairness, we trained BoQ, SALAD, and SALAD+CM for 30 epochs on the GSV-Cities dataset until convergence. As shown in Table 6, most methods benefit significantly from higher feature dimensions. In contrast, our method shows only a moderate improvement when increasing the dimension from low (Table 1 and Table 2) to high, possibly because our dimensionality control is achieved through a simple MLP, whereas other methods involve additional internal projection layers.

Nevertheless, our method still delivers outstanding performance, achieving the best results in both R@1 and R@5 metrics.

C.2 Fine-tuning Strategies

To assess the effectiveness of our framework in accommodating different fine-tuning strategies, we evaluated MulConV, the method used in our work for adapting DINOv2 features, against PEFT-based approaches proposed in SelaVPR [21] and EDTformer [16]. Table 7 presents retrieval performance across multiple datasets, including Tokyo 24/7, MSLS-Val, Pitts30k, SF-XL-v1, and SF-XL-Occlusion.

Method	Tokyo247	MSLS-val	Pitts30k	SF-XL-v1	SF-XL-Occlusion
SelaVPR	90.9 / 96.1	86.4 / 93.6	89.9 / 95.8	75.3 / 84.3	40.5 / 54.1
EDTformer	87.3 / 93.7	85.5 / 93.8	89.5 / 95.7	76.4 / 82.7	38.8 / 52.0
MulConV (Ours)	92.1 / 96.5	89.2 / 95.1	90.9 / 96.4	80.8 / 86.4	47.4 / 65.8

Table 7: **Comparison of different fine-tuning strategies within our framework.** Metrics are R@1 / R@5.

The results demonstrate that all fine-tuning strategies achieve competitive performance, indicating that our framework effectively leverages pretrained representations. Among the evaluated methods, MulConV consistently delivers the highest retrieval accuracy across all datasets, particularly under challenging conditions with occlusions or extreme viewpoint variations. This superior performance motivated our choice of MulConV as the fine-tuning strategy in our framework. These findings highlight both the flexibility and robustness of our approach, showing that it can accommodate various adaptation methods while benefiting most from MulConV.

C.3 Comparison of Different Backbones

To investigate the impact of different backbone architectures on retrieval performance, we conduct experiments using VGG16, ResNet50, and DINOv2 as feature extractors under a consistent training protocol (with $K = 3$ and no differential cropping). The results are shown in Table 8.

Backbone	**Params.**	**Flops.**	**SF-XL-testv1**	
			R@1	**R@5**
VGG16	15.0m	77.5b	61.5	70.8
ResNet50	24.6m	21.3b	73.9	81.4
DINOv2	100.9m	122.8b	**77.0**	**84.6**

Table 8: **Comparisons of various backbone.** Train under different backbones when K=3 on origin dataset without multi-angle cropping. For VGG16, only the parameters of the last layer are trained. For ResNet50, only the parameters beyond the third layer are trained. For DINOv2, only the adapter module is trained.

Despite having significantly fewer FLOPs than VGG16, ResNet50 achieves much better performance (R@1 of 73.9 vs. 61.5), highlighting the advantage of deeper residual connections and stronger feature representations. DINOv2, a vision transformer pretrained with self-supervised learning, achieves the best retrieval accuracy (R@1 of 77.0 and R@5 of 84.6), even though only a small adapter is trained on top. This confirms the strong generalization and representational capacity of DINOv2 features, making it a suitable backbone for downstream place recognition tasks.

These findings support our choice of using DINOv2 in the main experiments, striking a good balance between high performance and efficient finetuning.

D Discussion on EigenPlaces' Performance on SF-XL-testv1

Although our method generally outperforms existing approaches, it shows a slight drop on SF-XL-testv1, where EigenPlaces performs marginally better.

This arises from their methodological difference: EigenPlaces, being geometry-driven, groups views of the same focal point from different directions, thus enforcing a strong viewpoint-invariant prior that aligns well with SF-XL's panorama-cropped, multi-view structure.

In contrast, our approach clusters images purely in visual feature space. While it lacks explicit geometric constraints, it learns semantically and spatially coherent clusters that generalize more flexibly to complex scenes with occlusion, clutter, or fine-grained variations.

To verify the above observation and better understand the performance gap on SF-XL-testv1, we conducted a quantitative analysis to measure the degree of viewpoint invariance across different class construction strategies. Specifically, we sampled approximately 9k UTM grids (about 1M images) and applied k-means clustering using three methods: CosPlace, EigenPlaces, and ours. For each grid, we computed the feature distances between clusters with opposing headings (0° vs. 180°), as illustrated in Fig. 7.

The average inter-cluster distances were CosPlace: 1.1614, EigenPlaces: 1.0759, and Ours: 1.1247. These results quantitatively confirm that EigenPlaces exhibits the strongest viewpoint invariance, which explains its superior performance on benchmarks characterized by extreme viewpoint changes such as SF-XL-testv1. Conversely, CosPlace shows the largest inter-cluster distance and correspondingly lower performance, further validating that this metric meaningfully reflects a model's sensitivity

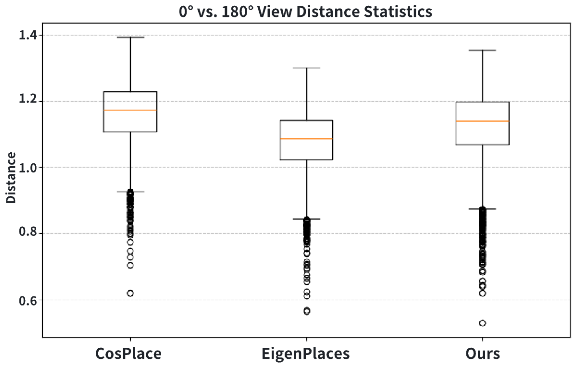

Figure 7: **Comparison of viewpoint invariance.** Box plot of inter-cluster distances (0° vs. 180°) for CosPlace, EigenPlaces, and Ours. Smaller values indicate stronger viewpoint invariance.

to viewpoint variation. Our method lies between the two, achieving a balanced trade-off—less rigid viewpoint invariance but greater robustness to occlusion, scene clutter, and appearance ambiguity.