Pose-Aware Self-Supervised Learning with Viewpoint Trajectory Regularization

Jiayun Wang[1]
[1]UC Berkeley
peterwg@berkeley.edu

Yubei Chen[2]
[2]UC Davis
ybchen@ucdavis.edu

Stella X. Yu[1,3]
[3]University of Michigan
stellayu@umich.edu

Abstract. Learning visual features from unlabeled images has proven successful for semantic categorization, often by mapping different *views* of the same object to the same feature to achieve recognition invariance. However, visual recognition involves not only identifying *what* an object is but also understanding *how* it is presented. For example, seeing a car from the side versus head-on is crucial for deciding whether to stay put or jump out of the way. While unsupervised feature learning for downstream viewpoint reasoning is important, it remains under-explored, partly due to the lack of a standardized evaluation method and benchmarks.
We introduce a new dataset of adjacent image triplets obtained from a viewpoint trajectory, without any semantic or pose labels. We benchmark both semantic classification and pose estimation accuracies on the same visual feature. Additionally, we propose a viewpoint trajectory regularization loss for learning features from unlabeled image triplets. Our experiments demonstrate that this approach helps develop a visual representation that encodes object identity and organizes objects by their poses, retaining semantic classification accuracy while achieving emergent global pose awareness and better generalization to novel objects. Our dataset and code are available at http://pwang.pw/trajSSL/.

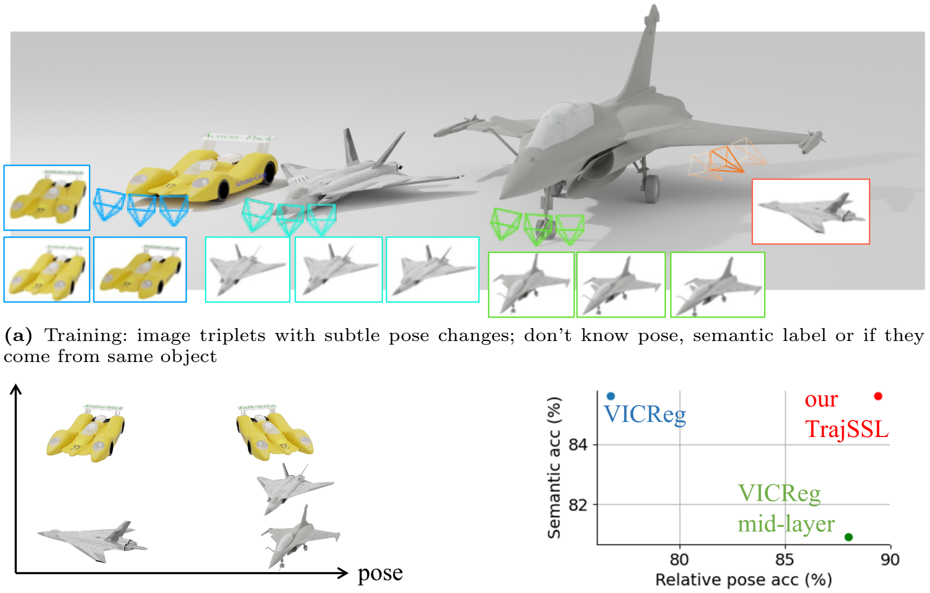
Keywords: Self-Supervised Learning · Pose Estimation · Trajectory

1 Introduction

Learning visual features from unlabeled images has proven successful for semantic categorization. Compared to supervised feature learning, self-supervised learning (SSL) can discover data patterns without labels [4,10,31,56], improve the performance of large-scale vision and language models [1,3], remain highly flexible [43] and generalizable to real-world data [44,53].

SSL methods so far have focused on coarse-grained recognition, by mapping different *views* of the same object to the same feature to achieve recognition invariance [4,10,31,56]. Consequently, both task-specific [53] and foundational models [22] are poor at recognizing objects with unseen or rare poses. Most data collections do not evenly cover the full range of object poses, while the training data is pivotal for robust performance [39,47]. Lacking pose awareness makes SSL methods worse at generalizing to novel poses.

However, visual recognition involves not only identifying *what an object is* but also understanding *how it is presented*. For example, seeing a car from the



(a) Training: image triplets with subtle pose changes; don't know pose, semantic label or if they come from same object

(b) Expected Learned Representation

(c) Results

Fig. 1: Our goal is to capture two aspects of object recognition through SSL: *what the object is* and *how the object is presented*. While the former has been well studied [4, 10], the latter is rarely understood. We learn SSL representations that not only capture object identity but also pose. **(a)** The training data are image triplets with subtle viewpoint changes of an object. The object identity, semantics and pose are unknown. **(b)** The learned representations are expected to discriminate different object semantics and poses and to achieve high accuracies for both semantic classification and pose estimation. Notably, we expect to understand *global pose* from *local pose* changes. **(c)** Our approach improves pose estimation accuracy over existing methods [4] by encouraging images with similar poses to form smooth trajectories in the representation space.

side versus head-on is crucial for deciding whether to stay put or jump out of the way. While unsupervised feature learning for downstream viewpoint reasoning is important, SSL is evaluated mostly on semantic tasks, e.g. classification and detection, and its effectiveness for pose-aware representation learning remains under-explored [18], without a standardized evaluation method.

We extend the concept of SSL to visual recognition beyond coarse semantic categorization. We aim at learning a pose-aware visual representation from naturally available visual data, so that it can support down-stream semantic classification and viewpoint estimation (Fig.1).

We first introduce a new object-centric dataset of adjacent image triplets obtained from a viewpoint trajectory, without any semantic or pose labels. Such a data acquisition scheme is most natural: 1) In human vision, even during fixation at a stationary object, our eyes make continuous and minute movements, including tremor, drift, and microsaccade; 2) In robotic vision, as the robot moves around in the environment, it captures adjacent images of the same object from a smooth viewpoint trajectory. We generate synthetic image triplets of various objects with slight camera pose changes.

We then benchmark semantic classification and pose estimation on the feature learned from unlabeled image triplets, for seen and unseen objects, as desirable for video analysis [28], robotics [21] and world-models [48]. Our benchmark precludes the use of semantic or pose labels during training and encompasses both semantic classification and pose estimation tasks during evaluation, unlike existing settings [26, 38] which allow training SSL with pose labels.

We include both absolute and relative pose estimation tasks. The former is useful for testing how well SSL learns a global pose from adjacent poses, whereas the latter is useful for testing how well SSL generalizes to out-of-domain poses. Without defining category-specific canonical poses, SSL can be flexibly evaluated on out-of-domain data and unseen semantic categories.

We benchmark existing SSL methods with a ResNet backbone. We discover that intermediate-layer features outperform later-layers by absolute 10-20% gains in pose estimation, at reduced semantic classification accuracies. This result is not surprising, as the last-layer feature under the SSL objective becomes more invariant to the object pose and tuned to the semantic category.

We further improve the performance by proposing a viewpoint trajectory regularization loss on intermediate features. Inspired by slow feature analysis [16, 25,27,55], we encourage adjacent views in the triplet to form a smooth trajectory in the feature space, implemented with a simple local linearity assumption.

Our simple approach leads to an additional 4% gain in pose estimation without affecting semantic classification. It is also more effective at out-of-domain generalization and on a real-world rotating-car benchmark Carvana [50].

Our work has three main contributions. **1)** We introduce a new dataset of unlabeled image triplets and a new SSL benchmark for both semantic classification and pose estimation. **2)** We propose a novel viewpoint trajectory regularization loss on intermediate features. **3)** We demonstrate that our simple approach helps develop a visual representation that encodes object identity and organizes objects by pose, retaining semantic classification accuracy while achieving emergent global pose awareness and better generalization to novel objects.

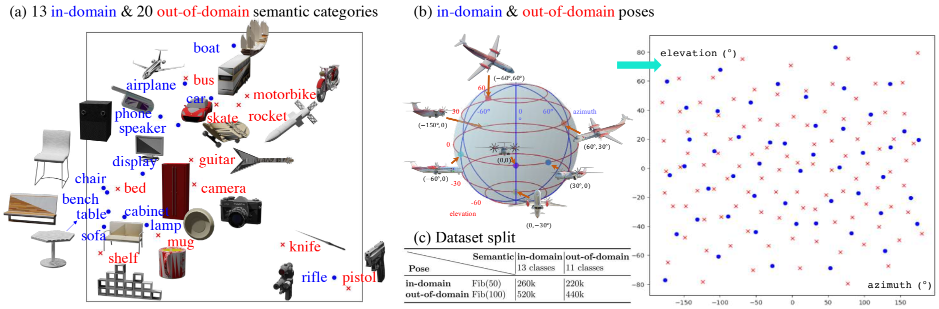
2 Related Works

Self-Supervised Learning for Semantic Downstream Tasks. There are predominantly two SSL approaches: contrastive and non-contrastive. Contrastive methods, grounded in the InfoNCE criterion [42], include [10,11,14,31]. A notable variant is clustering-based contrastive learning [6, 7, 43], which shifts focus from individual samples to cluster centroids. Non-contrastive approaches [4, 5, 12, 23, 29], on the other hand, aim to align embeddings of positive pairs, similar to contrastive learning, but with strategies to prevent representational collapse. Yet, they primarily focus on semantic tasks like semantic classification, leaving geometric tasks such as pose estimation underexplored. We bridge the gap by also providing the benchmark for geometric downstream tasks. We refer to works above as *invariant SSLs* as they learn representations invariant to object pose.

Geometry-Aware Self-Supervised Learning. In the quest for geometry-aware SSL, a prevalent method is to learn equivariant representations. Past research has utilized autoencoders, including transforming autoencoders [33], Homeomorphic VAEs [24], or [54]. Recently, EquiMod [20] and SEN [45] have introduced predictors that enable reconstruction-free representation manipulation in the latent space. Another novel approach is learning equivariant representations without prior knowledge of transformation groups, as explored in [49].

Perhaps the most relevant work to us is SIE [26], where they first provide an approach and dataset to evaluate equivariant representation learning via a downstream task of rotation matrix prediction. A critical distinction is that SIE utilizes ground-truth pose labels during training, aligning more with geometry-supervised methods, whereas our approach strictly prohibits the use of any geometric or semantic labels in SSL training. We also differ from SIE in that we evaluate pose estimation performance for out-of-domain data with relative pose. We summarize different methods in Table 3 in supplementary.

Pose Estimation. We adopt pose estimation as a task to evaluate geometric representations, given its fundamental role in many geometry-aware recognition tasks [35,41]. The object pose remains ambiguous unless a canonical pose is defined. However, defining the canonical pose can be challenging for a set of objects with different semantic classes. It is also hard to define a general canonical pose for all categories due to the difficulty of aligning two classes (e.g. airplanes and boats). Relative pose estimation can be used to eliminate the need for canonical pose. Specifically, there are two pose estimation evaluation methods: **1) Absolute pose estimation from a single image** is only well-defined if a canonical pose (or canonical coordinate system) exists. Previous work on single-view pose estimation is therefore class-specific. For a fixed set of categories, they define canonical coordinate systems class-by-class with a prior [9,34,36,37] or learned features [52]. On the contrary, we achieve class-agnostic absolute pose estimation with k nearest neighbor retrieval (k-NN): we first ask the model to find the most k similar representations and assume they are all from the same semantic category (e.g. airplanes). As the pose labels of all instances of the same semantic category follow a consistent predefined canonical coordinate system, then the predicted pose label with k-NN should also consistently follow the canonical coordinate system. **2) Relative pose estimation from a pair of images** avoids class-specific canonical coordinate system by assuming the first image defines a canonical pose, and thus predicting the relative pose of the second image compared to the first image is not ambiguous. RelPose and its variants [40,58] describe a data-driven method for inferring the relative pose given an image pair, and we adopt their setting for our class-agnostic relative pose estimation.


(a) 13 in-domain & 20 out-of-domain semantic categories

(b) in-domain & out-of-domain poses

(c) Dataset split

Pose \ Semantic		in-domain 13 classes	out-of-domain 11 classes
in-domain	Fib(50)	260k	220k
out-of-domain	Fib(100)	520k	440k

Fig. 2: Our benchmark dataset contains rendered images from ShapeNet [8]. **(a)** For semantics, we use non-overlapping 13 in-domain semantic categories and 11 out-of-domain categories. On the left, we project in-domain and out-of-domain semantic classes with PCA-projected Word2Vec [17] and show a representative object with $(15°, 15°)$ pose. **(b)** For pose, we adopt *absolute* and *relative* pose estimation as geometric evaluations. Notably, relative pose enables SSL's generalizability test on out-of-domain data as it eliminates the need for category-specific canonical pose. The (camera) pose is defined as the spherical coordinate (azimuth, elevation) of the camera position. We render objects from n unique camera angles, uniformly distributed across the viewing sphere S^2, utilizing a Fibonacci sphere distribution [2], denoted as Fib(n). We use Fib(50) as in-domain training and Fib(100) for out-of-domain evaluations. **(c)** Number of images for in-domain and out-of-domain sets. Data with both in-domain semantics and pose is used for training, while the rest is for evaluation.

3 A Benchmark for SSL Geometric Representations

3.1 The Problem Setting

We propose a benchmark that evaluates the SSL semantic and geometric representation quality. The SSL operates without ground-truth semantic or pose labels during training, aiming to develop representations that are aware of both the semantics and geometry of the input image. Key principles for the SSL benchmark include: **1) Training Phase:** SSL is trained purely on images, without any semantic or pose labels. This ensures that all learned information is derived directly from the image itself. **2) Evaluation Phase:** SSL should learn representations that encode both semantic and geometric information. Using different representations from the model for different tasks is fine.

Given the nature of SSL, pose labels are explicitly excluded during training to align with the principle of learning without labels. One may argue that in-plane image rotations (as suggested by [18]) could offer pseudo "pose labels", but more complex manipulations like 3D rotations are generally unfeasible. We thus strictly avoid any labels during training. In the evaluation phase, SSL must learn both semantic and geometric representations from data, since both elements and their interplay are essential for a comprehensive understanding of the data and could benefit the overall learning process. One challenge lies in estimating the pose of an out-of-domain image, as the canonical pose is not defined. We thus introduce relative pose estimation as the metric, with details as follows.

3.2 Data and Evaluation Metrics

Our benchmark provides data generation, downstream tasks and evaluation configurations, to evaluate SSL's capacity to capture geometric and semantic information.For data generation and configuration methods, without loss of generalizability, we use 3D meshes from ShapeNet [8] to generate images of varied objects in diverse 3D poses as the dataset used empirically in this work. For the geometric task, we adopt a fundamental one, pose estimation of object-centric images. We consider both *absolute* and *relative* 3D pose estimation tasks to enable evaluations on in-domain and out-of-domain data, where we also provide a dataset-splitting configuration. Compared to existing datasets with similar purposes [26, 59], ours enables out-of-domain evaluation and the generation method leads to a complete and even pose coverage. Shadows are also not rendered to avoid unintended ground-truth pose information leakage. We provide a detailed comparison with such datasets in supplementary.

Pose. We make source 3D objects of the same semantic class aligned and fixed for image rendering. The pose is defined as the camera pose. Specifically, we make cameras all reside on a unit S^2 sphere (Fig.2b) with rendering configurations of look-at view transform with up vector $(0, 1, 0)$ and translation vector $(0, 0, 1)$ (following PyTorch3D's convention [46]). Camera poses are represented as (azimuth, elevation) pairs, the spherical coordinates of camera positions. We define the category-specific canonical pose to ensure no *absolute pose* ambiguity for in-domain data as objects of each semantic category are aligned.

Relative Pose eliminates the necessity of canonical pose by considering two views of an object with pose $\mathbf{p_1}, \mathbf{p_2}$, and is defined as $\mathbf{\Delta p} = \mathbf{p_2} - \mathbf{p_1}$, the pose difference from view 2 to view 1. Introducing relative pose ensures the SSL generalizability evaluation on out-of-domain images where canonical poses are not tractable. This differs from the previous setting [26] where out-of-domain data cannot be considered with only absolute pose estimation.

Pose Sampling. For uniform camera coverage of the whole viewing sphere, for each object, we use Fibonacci lattices [2, 30], placing n cameras at each lattice point to render n views, denoted as Fib(n). We render in-domain poses using Fib(50) and out-of-domain poses with Fib(100), rotating Fib(100) to avoid pose overlap. Fig.2b depicts rendered images with fixed pose.

Dataset Split. We divide the dataset for in-domain and out-of-domain parts (Fig.2c). For the pose, we use non-overlapping Fib(50) and Fib(100) for in-domain and out-of-domain poses as mentioned before. For semantic categories, we use 13 object classes (e.g. airplane, car, watercraft, etc.) as in-domain data, and 11 object classes (e.g. bed, guitar, rocket, etc.) as out-of-domain data. There is no overlapping for the two sets. For each semantic category, we render 400 different objects, with 320 for unsupervised training (or probe training) and 80 for testing. For simplicity, our experiments focus on cases where either pose or semantics are out-of-domain, not both.

Downstream Tasks and Evaluation. We follow previous benchmarks for semantic classification. To evaluate geometric representation, our benchmark includes the following downstream task configurations with ShapeNet [8] as an

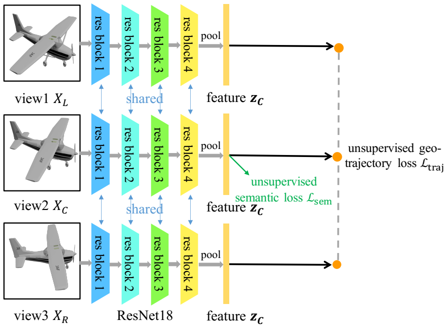

res block 1
res block 2
res block 3
res block 4
pool
view1 X_L
shared
feature $\boldsymbol{z_C}$
view2 X_C
unsupervised geo-trajectory loss $\mathcal{L}_{\text{traj}}$
unsupervised semantic loss $\mathcal{L}_{\text{sem}}$
view3 X_R
ResNet18

Fig. 3: In addition to an unsupervised semantic loss $\mathcal{L}_{\text{sem}}$ which is commonly used in SSL, we add a trajectory loss $\mathcal{L}_{\texttt{traj}}$ (Eqn.3) to enhance geometric representation. $\mathcal{L}_{\text{sem}}$ always follows baseline settings, which is applied post-projector for SimCLR, for example. $\mathcal{L}_{\texttt{traj}}$ always operates on the pooled feature z. For pose evaluation, we allow representations from different layers and find that mid-layer representations like "res block3" give pose estimation gain.

example (Fig.2): *1) In-Domain: Absolute Pose.* Utilizing 13 in-domain semantic categories and poses from Fib(50), we assess absolute pose through nearest neighbor retrieval. *2) In-Domain: Relative Pose.* This task maintains the same data setting as the in-domain absolute pose, while the distinction lies in the training method. After unsupervised training, we employ a simple probe to train on 80% of the instances' frozen representations for relative pose estimation, with the remaining 20% used for performance evaluation. *3) Out-of-Domain: Unseen Poses.* We work with the 13 in-domain semantic categories but with unseen poses from Fib(100). We only evaluate relative pose estimation performance with a simple probe for faster inference speed. *4) Out-of-Domain: Unseen Semantic Categories.* This scenario involves 11 unseen semantic categories paired with in-domain poses from Fib(50). Similarly, we evaluate relative pose estimation performance.

4 Enhancing Geometric Representation Learning

4.1 Mid-Layer Representation for Evaluation

We explore the feasibility of using different layers of the backbone to predict pose. This consideration stems from the understanding that geometric tasks are typically mid-level vision tasks, whereas semantic tasks align with high-level ones. Additionally, unlike whole-image embedding which is approximately the average of local patch embeddings [15], mid-level features are local embeddings that could capture mid-level visual cues like pose. Mid-level features can thus be considered as a combination of patch embeddings that enhance pose estimation. We focus on whether using representations from mid-layers, such as the "res block3" or "res block4" layers (referenced in Fig.3), enhances pose estimation performance. For simplicity, we refer to "res block3" as "conv3" hereafter.

Empirically, our results show a significant improvement in pose estimation, with gains ranging from 10%-20% when using mid-layer representations (detailed in Section 5.4). Further, as a verification of the similarity between mid-layer representations and patch embeddings, we concatenate embeddings of local image patches and observe a similar performance to "conv4" embedding (87% vs 88%, details in supplementary). A common challenge with mid-layer representations is

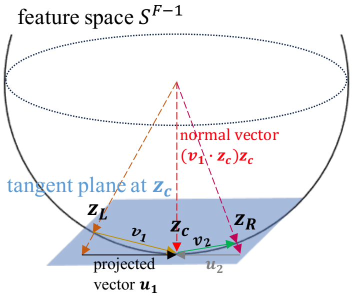


Fig. 4: We enforce representations of adjacent views of an object, $\mathbf{z_L}, \mathbf{z_C}, \mathbf{z_R}$, to form a geodesic trajectory. **Upper:** $\mathbf{z}$ resides on a unit hypersphere. The objective is to map the difference vectors $\mathbf{v_1} = \mathbf{z_C} - \mathbf{z_L}$ and $\mathbf{v_2} = \mathbf{z_R} - \mathbf{z_C}$ onto $\mathbf{z_C}$'s tangent plane, optimizing for maximal cosine similarity to achieve a linear trajectory on that plane. **Lower:** Projected vector $\mathbf{u}$ is computed by deducting the normal component $\mathbf{z_C}$ from the difference vector $\mathbf{v}$.

their high dimensionality, primarily due to large spatial sizes. This high dimensionality can lead to inefficiencies during inference and storage. We demonstrate in Section B that high-dimensional mid-layer representations can be effectively compressed with minimal accuracy drop, thereby enhancing overall efficiency.

4.2 Trajectory Regularization

Given an image X of an object with pose $\mathbf{p}$, we feed it to an encoder f_θ to obtain a representation $\mathbf{z} = f_\theta(X)$, which is used for both semantic and geometric tasks.

Invariant SSL refer to methods [4,10,31] that generate representations that are invariant to data augmentations (e.g., random crops), which sometimes include geometric augmentations, due to the primary focus on semantic representations. For an image X, invariant SSLs create two augmented variants, X_{T_1} and X_{T_2}, which are then fed into the encoder f_θ for two respective representations, $\mathbf{z_{T_1}}$ and $\mathbf{z_{T_2}}$. The invariant loss, or unsupervised semantic loss is method-dependent and can be denoted as $\mathcal{L}_{\text{sem}}(\mathbf{z_{T_1}}, \mathbf{z_{T_2}})$. Despite their focus on semantic information, we evaluate such invariant SSL representations for predicting pose $\hat{\mathbf{p}}$ of image X, considering that pose information might be encoded within $\mathbf{z}$.

Trajectory-Regularized SSL. We aim to enhance the SSL geometric representation quality by leveraging a natural prior: representations of objects with incremental pose changes should form a smooth, low-curvature path in the representation space. This leads us to promote a locally linear trajectory for representations corresponding to slight pose variations. Linear trajectory requires small camera pose changes only and does not violate the SSL setting.

Consider a triplet of images $\{X_L, X_C, X_R\}$ from a sequence with respective poses $\mathbf{p_L}, \mathbf{p_C}, \mathbf{p_R}$ forming a trajectory, where pose changes are subtle. That is, $\{X_L, X_C, X_R\}$ form an adjacent pose triplet. The encoded representations $\mathbf{z_L}, \mathbf{z_C}, \mathbf{z_R}$ are normalized and residing on a unit hypersphere (i.e., $|\mathbf{z}|_2 = 1$). Our goal is to align these points along a geodesic trajectory on the hypersphere. This is achieved by projecting the difference vectors between representations onto the tangent plane at $\mathbf{z_C}$, thereby enforcing a linear trajectory (Fig.4).

The difference of two representations with adjacent poses is $\mathbf{v_1} = \mathbf{z_C} - \mathbf{z_L}, \mathbf{v_2} = \mathbf{z_L} - \mathbf{z_C}$. These vectors are projected on the tangent space at $\mathbf{z_C}$:

$$\mathbf{u_1} = \mathbf{v_1} - (\mathbf{v_1} \cdot \mathbf{z_C})\mathbf{z_C} \tag{1}$$

$$\mathbf{u_2} = \mathbf{v_2} - (\mathbf{v_2} \cdot \mathbf{z_C})\mathbf{z_C} \tag{2}$$

We then maximize the cosine similarity between $\mathbf{u_1}$ and $\mathbf{u_2}$ to enforce linearity in the trajectory. The trajectory loss, or pose loss, is defined as:

$$\mathcal{L}_{\text{traj}}(\mathbf{z_L}, \mathbf{z_C}, \mathbf{z_R}) = -\frac{\mathbf{u_1} \cdot \mathbf{u_2}}{\|\mathbf{u_1}\|\|\mathbf{u_2}\|} \tag{3}$$

Semantic loss $\mathcal{L}_{\text{sem}}$ is also incorporated for semantic representation capacity. We apply augmentations to X_C to generate X_{T_1} and X_{T_2}, then apply semantic loss on their representations $\mathcal{L}_{\text{sem}}(\mathbf{z_{T_1}}, \mathbf{z_{T_2}})$. Our total loss combines both semantic and pose losses (Fig.3):

$$\mathcal{L} = \mathcal{L}_{\text{sem}}(\mathbf{z_{T_1}}, \mathbf{z_{T_2}}) + \lambda \mathcal{L}_{\text{traj}}(\mathbf{z_L}, \mathbf{z_C}, \mathbf{z_R}) \tag{4}$$

where weight λ balances the trajectory loss. We always apply the trajectory loss $\mathcal{L}_{\text{traj}}$ (as per Eqn.3) at the pooled feature layer z, as empirically altering the layer in $\mathcal{L}_{\text{traj}}$ impacts downstream task performance by only about 1%.

5 Experiments

We first discuss the training and evaluation protocols, and then report and compare evaluation performance when using the last and mid-feature layer as the representation. We conclude the section with representation visualizations.

5.1 Training Protocols

Our experimental framework adopts the common two-stage approach used in SSL. Supervised baselines are included as references. The first stage is unsupervised pretraining (or supervised training), where the model is fed with training data. The specific training protocols of fully-supervised, geometry-supervised and self-supervised methods are method-dependent. The second stage is evaluation, which includes directly evaluating the learned representation on downstream tasks (with the nearest neighbor) and simple probes trained on frozen representations for downstream tasks. The second evaluation stage is the same for all methods for fairness (details in Section 5.2). We mainly consider three baselines: fully-supervised, geometry-supervised, invariant SSL. We discuss the training settings of each method below.

Fully-Supervised Learning. We provide supervised baselines to establish an upper bound for in-domain performance (Table 1, first row). Separate models for semantic classification and pose estimation are trained with corresponding ground-truth labels to prevent task interference.

Geometry-Supervised Learning. Following methods like [18,26,38], baselines are trained on ground-truth pose labels but not semantic labels during training

stage (Table 1, second row). Specifically, we replicate the AugSelf [38] setting, combining an unsupervised semantic loss $\mathcal{L}_{\text{sem}}$ with a cross-entropy loss for pose labels. The results are for reference and not a direct or fair comparison to SSLs.
Invariant Self-Supervised Learning. We consider two state-of-the-art SSL methods, VICReg [4] and SimCLR [10]. Images of the same object with different poses are treated as distinct samples. We follow their training settings and use standard data augmentation (e.g. random crop and color jittering).
Trajectory-Regularized Self-Supervised Learning. We add the trajectory loss $\mathcal{L}_{\text{traj}}$ to the invariant SSL [4, 10]. As mentioned earlier, we assume image triplets from a sequence with small relative pose changes are available. We implement as follows: during training, for an image X_C with pose $\mathbf{p_C}$, we randomly select an adjacent left image X_L with pose $\mathbf{p_L}$. Using slerp [51], we obtain the right pose p_R such that $\mathbf{p_R} - \mathbf{p_C} = \mathbf{p_C} - \mathbf{p_L}$, and render the right image X_R. The image triplet $\{X_L, X_C, X_R\}$ can now be used to obtain the trajectory loss. Importantly, no additional transformations like random cropping are applied to X_L, X_C, X_R to preserve geometric information. Our method still works for non-equidistant poses, i.e. $\mathbf{p_R} - \mathbf{p_C} \neq \mathbf{p_C} - \mathbf{p_L}$, with details in supplementary.
Shared Protocols. All methods utilize a ResNet-18 [32] as the backbone encoder[1]. Training is consistent across models, spanning 300 epochs using the LARS optimizer [57] with a learning rate of 0.3 and weight decay of 10^{-4}.

5.2 Evaluation Protocols

Semantic Classification. We evaluate with a linear classification on top of the frozen representation from the feature layer with dimension 512.
Pose Estimation. As a comprehensive evaluation, we consider both absolute and relative pose estimation tasks: **1) Absolute pose estimation.** We employ a weighted k-nearest neighbor classifier as used in [56] on the representations from the feature layer. **2) Relative pose estimation.** We obtain feature-layer representations z_1, z_2 from two different views of an object. These representations are concatenated (resulting in a 1024-dim feature), and a simple probe of a two-layer perceptron with a hidden dimension of 1024 is used to predict the relative pose. Relative pose estimation is more computationally efficient but is generally harder as it relies on only two views for inference. We consider in-domain and out-of-domain scenarios for pose estimation and only report relative pose performance to avoid redundancy (as mentioned in Section 3.2).

5.3 Evaluation on Last Feature-Layer

We report the semantic classification and pose estimation performance of different methods in Table 1. For geometry-supervised and SSL methods, the same feature-layer representation z is used for both geometric and semantic tasks. We aim to understand:**1)** if adding trajectory loss $\mathcal{L}_{\text{traj}}$ (Eqn.3) helps pose estimation, and **2)** what is the gap between SSL and supervised methods.

[1] Our method also works with other model architecture. See supplementary for details.

Table 1: The proposed trajectory loss $\mathcal{L}_{\text{traj}}$ leads to pose estimation gain without harming semantic classification accuracy. Specifically, SSL gives comparable or marginally superior results than supervised methods for out-of-domain and real data. Feature-layer representation z is used for both semantic and pose estimation.

Acc. (%)	**In-Domain**					**Out-of-Domain Pose Est.**				**Real Photos**	
	sem. cls.	**abs. pos**	**our gain**	**rel. pose**	**our gain**	**unseen sem.**	**our gain**	**unseen pose**	**our gain**	**Cars** [50]	**our gain**
Fully-Sup. [a]	86.4	92.2		86.1		61.3		77.4		88.5	
Geometry-Sup.	85.4	89.8		83.8		61.4		77.6		87.9	
Fully-unsupervised methods											
VICRreg [4]	85.6	84.3		76.7		59.6		73.1		88.7	
VICRreg+traj.	85.6	**87.8**	3.5	**80.5**	3.8	**62.7**	3.1	**77.5**	4.4	**91.7**	3.0
SimCLR [10]	85.9	84.8		77.3		58.1		68.5		89.0	
SimCLR+traj.	**86.0**	86.4	1.6	79.5	2.2	61.3	3.2	71.0	2.5	91.5	2.5
SimSiam [13]	85.4	84.9		77.4		57.8		68.1		88.8	
SimSiam+traj.	85.5	87.2	2.3	79.5	2.1	61.0	3.2	70.8	2.7	91.2	2.4

[a] We train two separate supervised models for semantic classification and pose estimation, as a supervised multi-task model yields worse results than specialized, separate models.

Semantic Classification. All methods have a similar semantic classification accuracy (85-86%). SSL accuracies are close to the supervised upper bound. Also, adding the trajectory regularization loss $\mathcal{L}_{\text{traj}}$ leads to no accuracy loss for semantic classification, indicating that geometric representation is learned without harming semantic tasks.

In-Domain Pose Estimation. Adding the trajectory regularization yields up to 4% performance gain, although there is a performance gap between SSL methods and supervised methods. Specifically, we consider two evaluation methods: absolute pose with k-NN and relative pose with simple probe. For the absolute pose estimation, adding the proposed trajectory loss leads to 4% gain for VICReg and 2% gain for SimCLR and SimSiam. For the relative pose estimation, adding the proposed trajectory loss also leads to 4% gain for VICReg and 2% gain for SimCLR and SimSiam. For both absolute and relative pose, SSL has a 2%-3% gap to geometry-supervised methods and 4%-5% gap to the supervised methods, which is expected as SSL takes no ground-truth pose labels.

Out-Of-Domain Pose Estimation. Trajectory loss $\mathcal{L}_{\text{traj}}$ yields up to 4% gain, and SSL methods are on par or even slightly outperform supervised methods on out-of-domain pose estimation. Specifically, for unseen poses, adding the proposed trajectory loss also leads to 3% gain for VICReg and 3% gain for SimCLR. SSL slightly outperforms supervised and geometry-supervised methods. For unseen categories, adding $\mathcal{L}_{\text{traj}}$ also leads to 4% gain for VICReg and 3% gain for SimCLR. SSL is on par with supervised and geometry-supervised methods.

Real Photos. Models trained on synthetic data can directly work on real data. Specifically, we directly evaluate models trained on the synthetic dataset [8] on a real photo dataset, Carvana [50], for pose estimation. We randomly use 80%

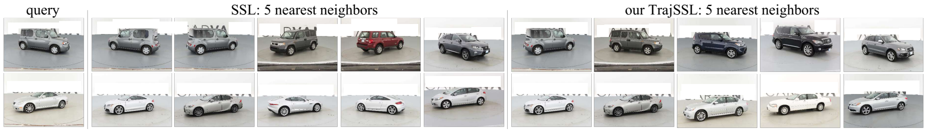


Fig. 5: Retrieval on a real rotating-car dataset [50]. Adding trajectory-regularization to a baseline SSL [4] leads to better retrievals with similar pose and appearance.

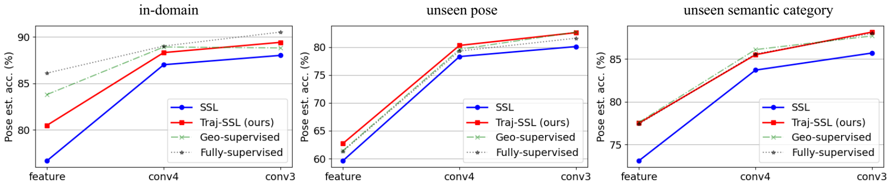


Fig. 6: Mid-layer representations improve pose estimation performance: 9% for in-domain data, 20% gain for out-of-domain poses and 11% gain for out-of-domain semantic classes. SSL's gap to supervised methods is also smaller for out-of-domain data.

Table 2: Using mid-layer "conv3" rather than last-layer "feature" for relative-pose-estimation downstream task improves accuracy: 9% for in-domain data and 20% for out-of-domain unseen poses.

Pose Acc. (%)	**In-Domain**			**Unseen Pose**			**Unseen Cat.**		
Representation layer	conv3	conv4	feat	conv3	conv4	feat	conv3	conv4	feat
Fully-Sup.	90.5	89.0	86.1	81.6	79.3	61.3	88.0	85.6	77.4
Geometry-Sup.	88.8	88.9	83.8	82.7	79.6	61.4	87.7	86.1	77.6
VICReg [4]	88.0	87.0	76.7	80.1	78.3	59.6	85.7	83.7	73.1
VICReg+traj.	**89.4**(↑9)	88.3	80.5	**82.6**(↑20)	80.3	62.7	**88.2**(↑11)	85.5	77.5

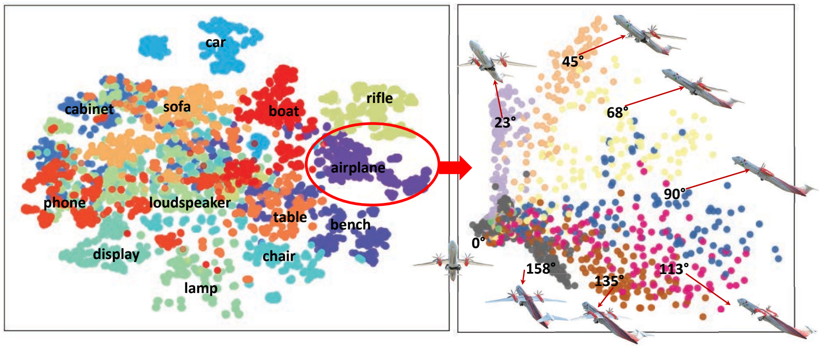
instances as the gallery and the rest as queries. The dataset contains 318 car instances, each of which has 16 views, leading to 5,088 car images in total. Adding $\mathcal{L}_{\text{traj}}$ also leads to 3%, 3% and 2% gain for VICReg, SimCLR and SimSiam. SSL slightly outperforms supervised methods. Retrieval results are in Fig.5.

In all, trajectory loss enhances in-domain and out-of-domain pose estimation. SSL is on par with supervised methods on out-of-domain pose estimation.

5.4 Evaluation on Mid-Layer Representations

During training, the trajectory loss $\mathcal{L}_{\text{traj}}$ (Eqn.3) is always constrained on feature layer z, as we find changing the layer used for $\mathcal{L}_{\text{traj}}$ only gives $\sim 1\%$ difference (Section D). After training the model, we could use different layers as the representation for downstream geometric tasks. We report relative pose estimation performance using representations of different layers of ResNet-18 [32] and find mid-layer "conv3" gives the best performance (Table 2 and Fig.6). For a fair comparison, all probes have the same parameter size.

In-Domain Pose Estimation. Using mid-layer representation "conv3" greatly enhances pose estimation performance over the last feature-layer. The gap is small compared with the second to the last layer "conv-4". Specifically, using



Multi-class representation

Airplane representation grouped by pose

Fig. 7: The joint semantic-pose embedding: Images are clustered by semantics; within each semantic cluster, images form mini-cluster by pose. **Left:** Representation z is grouped by different semantic categories. Images with the same semantic categories form clusters. **Right:** Zooming in one category, airplane, we visualize 200 instances with different poses. As the azimuth changes, their representation also forms a trajectory.

"conv3" layer as representation leads to 1% gain over "conv4" and 9% gain over 'feature" layer for VICReg with trajectory regularization. For baseline SSL and supervised methods, we also observe gain with mid-level representations.

Out-Of-Domain Pose Estimation. Using the mid-layer feature "conv3" enhances pose estimation performance on out-of-domain data, and the gap is larger for unseen poses. Specifically, for unseen poses, "conv3" layer leads to 2% gain over "conv4" and 20% gain over 'feature" layer for VICReg with trajectory regularization. For unseen semantic categories, "conv3" layer has 3% gain over "conv4" and 11% gain over 'feature" layer. For baseline SSL and supervised methods, we also observe gain with mid-level representations for out-of-domain data.

Using Mid-Layer for Semantic Classification. Empirically, we found that using "conv3" or "conv4" layer as representation for semantic classification does not make much difference (less than 1%).

Additional experimental results are in the supplementary. **1)** Mid-layer representations improve the pose estimation performance but can increase the computation burden due to their high dimensionality. We show that we can compress such representation to the same dimension as the last layer with minimal performance drop (Section B in the supplementary). **2)** We find that our method is robust to hyperparameters or settings including the layer to impose trajectory loss, the weight of trajectory loss and when images have non-equidistant poses. For different backbones, our method also maintains performance gain over baselines. Refer to Section D in the supplementary for more details.

5.5 Visualizations

Visualization of Different Semantic Categories. We present visualizations of the feature-layer z organized by different semantic categories (Fig.7). We observe that images within the same semantic categories are naturally grouped

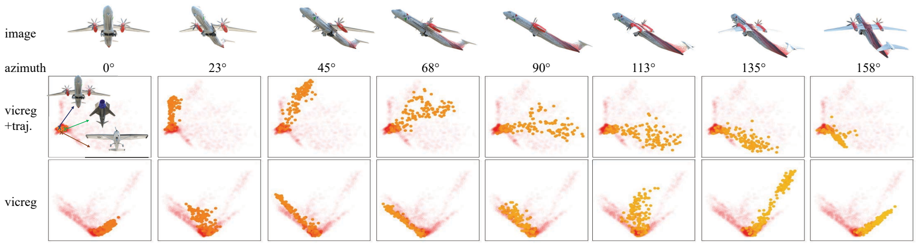


Fig. 8: PCA projection of embedding of renderings of multiple airplanes with pose changes, which demonstrates the improved representation of our method over baseline [4]. **Row 1**: Image with a pose for each column for visualizations. **Row 2-3**: The embedding is the same for each row, while each column highlights multiple airplanes with the same pose.In each sub-figure, different dots refer to different airplanes with the same pose. We observe as airplane poses change from $(0, 30°)$ to $(158°, 30°)$, their representations form a trajectory in the feature space. While the baseline method without trajectory loss can differentiate some views, it fails to form a trajectory, which could partially contribute to worse pose estimation performance.

together. For a specific category, airplane, we observe that as the pose varies, the representations also cluster together, with similar poses being closer.

Visualization of Different Poses. As object poses gradually change, their representations also form a trajectory (Fig.8). We visualize representations of 200 airplanes with poses ranging from $(0, 30°)$ to $(158°, 30°)$. These representations form a smooth trajectory. However, the baseline method without trajectory loss produces representations that can differentiate some views but may not form a coherent trajectory, which contributes to worse performance for pose estimation.

6 Summary

We introduce a new benchmark to evaluate geometric representations in self-supervised learning (SSL), which precludes semantic and pose labels during training. Based on the evaluation of state-of-the-art SSLs on the benchmark, we identified and implemented strategies that significantly enhance geometric representation capabilities. Notably, we observed a substantial improvement (10%-20%) in pose estimation performance by utilizing structured and mid-level representations. We further integrated an unsupervised trajectory regularization loss, which leads to an additional 4% enhancement. There are two major limitations: **1)** Our benchmark mainly uses synthetic data. **2)** While we utilize 3D pose estimation as our primary downstream task for evaluating geometric representations, the inclusion of more comprehensive geometric tasks, such as 6-DoF pose estimation or depth map prediction, could enrich the benchmark's scope and utility. That said, the proposed pose trajectory regularization is a general principle with the potential to benefit other geometric tasks. In conclusion, the proposed benchmark and approaches could augment SSL's proficiency and offer a comprehensive understanding of joint semantic-pose representation learning.

Acknowledgements. This project was supported, in part, by NSF 2215542 and NSF 2313151 to S. Yu. The authors thank Zezhou Cheng and Quentin Garrido for helpful discussions.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Alexa, M.: Super-fibonacci spirals: Fast, low-discrepancy sampling of so (3). In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8291–8300 (2022)
3. Bai, Y., Geng, X., Mangalam, K., Bar, A., Yuille, A.L., Darrell, T., Malik, J., Efros, A.A.: Sequential modeling enables scalable learning for large vision models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22861–22872 (2024)
4. Bardes, A., Ponce, J., LeCun, Y.: Vicreg: Variance-invariance-covariance regularization for self-supervised learning. arXiv preprint arXiv:2105.04906 (2021)
5. Bardes, A., Ponce, J., LeCun, Y.: Vicregl: Self-supervised learning of local visual features. Advances in Neural Information Processing Systems **35**, 8799–8810 (2022)
6. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European conference on computer vision (ECCV). pp. 132–149 (2018)
7. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
8. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
9. Chen, B., Chin, T.J., Klimavicius, M.: Occlusion-robust object pose estimation with holistic representation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2929–2939 (2022)
10. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
11. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
12. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
13. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021)
14. Chen*, X., Xie*, S., He, K.: An empirical study of training self-supervised vision transformers. arXiv preprint arXiv:2104.02057 (2021)
15. Chen, Y., Bardes, A., Li, Z., LeCun, Y.: Bag of image patch embedding behind the success of self-supervised learning. arXiv preprint arXiv:2206.08954 (2022)
16. Chen, Y., Paiton, D., Olshausen, B.: The sparse manifold transform. Advances in neural information processing systems **31** (2018)

17. Church, K.W.: Word2vec. Natural Language Engineering **23**(1), 155–162 (2017)
18. Dangovski, R., Jing, L., Loh, C., Han, S., Srivastava, A., Cheung, B., Agrawal, P., Soljačić, M.: Equivariant contrastive learning. arXiv preprint arXiv:2111.00899 (2021)
19. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153 (2023)
20. Devillers, A., Lefort, M.: Equimod: An equivariance module to improve self-supervised learning. arXiv preprint arXiv:2211.01244 (2022)
21. Du, G., Wang, K., Lian, S., Zhao, K.: Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. Artificial Intelligence Review **54**(3), 1677–1734 (2021)
22. El Banani, M., Raj, A., Maninis, K.K., Kar, A., Li, Y., Rubinstein, M., Sun, D., Guibas, L., Johnson, J., Jampani, V.: Probing the 3d awareness of visual foundation models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21795–21806 (2024)
23. Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. In: International Conference on Machine Learning. pp. 3015–3024. PMLR (2021)
24. Falorsi, L., De Haan, P., Davidson, T.R., De Cao, N., Weiler, M., Forré, P., Cohen, T.S.: Explorations in homeomorphic variational auto-encoding. arXiv preprint arXiv:1807.04689 (2018)
25. Földiák, P.: Learning invariance from transformation sequences. Neural computation **3**(2), 194–200 (1991)
26. Garrido, Q., Najman, L., Lecun, Y.: Self-supervised learning of split invariant equivariant representations. arXiv preprint arXiv:2302.10283 (2023)
27. Goroshin, R., Mathieu, M.F., LeCun, Y.: Learning to linearize under uncertainty. Advances in neural information processing systems **28** (2015)
28. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18995–19012 (2022)
29. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020)
30. Hardin, D.P., Michaels, T., Saff, E.B.: A comparison of popular point configurations on s2. arXiv preprint arXiv:1607.04590 (2016)
31. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
32. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
33. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21. pp. 44–51. Springer (2011)

34. Iwase, S., Liu, X., Khirodkar, R., Yokota, R., Kitani, K.M.: Repose: Fast 6d object pose refinement via deep texture rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3303–3312 (2021)
35. Kappler, D., Meier, F., Issac, J., Mainprice, J., Cifuentes, C.G., Wüthrich, M., Berenz, V., Schaal, S., Ratliff, N., Bohg, J.: Real-time perception meets reactive motion generation. IEEE Robotics and Automation Letters **3**(3), 1864–1871 (2018)
36. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE international conference on computer vision. pp. 1521–1529 (2017)
37. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE international conference on computer vision. pp. 2938–2946 (2015)
38. Lee, H., Lee, K., Lee, K., Lee, H., Shin, J.: Improving transferability of representations via augmentation-aware self-supervision. Advances in Neural Information Processing Systems **34**, 17710–17722 (2021)
39. Li, J., Fang, A., Smyrnis, G., Ivgi, M., Jordan, M., Gadre, S., Bansal, H., Guha, E., Keh, S., Arora, K., et al.: Datacomp-lm: In search of the next generation of training sets for language models. arXiv preprint arXiv:2406.11794 (2024)
40. Lin, A., Zhang, J.Y., Ramanan, D., Tulsiani, S.: Relpose++: Recovering 6d poses from sparse-view observations. arXiv preprint arXiv:2305.04926 (2023)
41. Macklin, M.: Warp: A high-performance python framework for gpu simulation and graphics. In: NVIDIA GPU Technology Conference (GTC) (2022)
42. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
43. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
44. Pantazis, O., Brostow, G.J., Jones, K.E., Mac Aodha, O.: Focus on the positives: Self-supervised learning for biodiversity monitoring. In: Proceedings of the IEEE/CVF International conference on computer vision. pp. 10583–10592 (2021)
45. Park, J.Y., Biza, O., Zhao, L., van de Meent, J.W., Walters, R.: Learning symmetric embeddings for equivariant world models. arXiv preprint arXiv:2204.11371 (2022)
46. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv:2007.08501 (2020)
47. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022)
48. Sergeant-Perthuis, G., Ruet, N., Rudrauf, D., Ognibene, D., Tisserand, Y.: Influence of the geometry of the world model on curiosity based exploration. arXiv preprint arXiv:2304.00188 (2023)
49. Shakerinava, M., Mondal, A.K., Ravanbakhsh, S.: Structuring representations using group invariants. Advances in Neural Information Processing Systems **35**, 34162–34174 (2022)
50. Shaler, B., Gill, D., Maggie, McDonald, M., Patricia, Cukierski, W.: Carvana image masking challenge. `https://kaggle.com/competitions/carvana-image-masking-challenge` (2017)
51. Shoemake, K.: Animating rotation with quaternion curves. In: Proceedings of the 12th annual conference on Computer graphics and interactive techniques. pp. 245–254 (1985)

52. Sun, W., Tagliasacchi, A., Deng, B., Sabour, S., Yazdani, S., Hinton, G.E., Yi, K.M.: Canonical capsules: Self-supervised capsules in canonical pose. Advances in Neural information processing systems **34**, 24993–25005 (2021)
53. Wang, J., Jeon, S., Yu, S.X., Zhang, X., Arora, H., Lou, Y.: Unsupervised scene sketch to photo synthesis. In: European Conference on Computer Vision. pp. 273–289. Springer (2022)
54. Winter, R., Bertolini, M., Le, T., Noé, F., Clevert, D.A.: Unsupervised learning of group invariant and equivariant representations. Advances in Neural Information Processing Systems **35**, 31942–31956 (2022)
55. Wiskott, L., Sejnowski, T.J.: Slow feature analysis: Unsupervised learning of invariances. Neural computation **14**(4), 715–770 (2002)
56. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3733–3742 (2018)
57. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
58. Zhang, J.Y., Ramanan, D., Tulsiani, S.: Relpose: Predicting probabilistic relative rotation for single objects in the wild. In: European Conference on Computer Vision. pp. 592–611. Springer (2022)
59. Zimmermann, R.S., Sharma, Y., Schneider, S., Bethge, M., Brendel, W.: Contrastive learning inverts the data generating process. In: International Conference on Machine Learning. pp. 12979–12990. PMLR (2021)

Table 3: Unlike supervised learning which requires labels in training, SSL uses neither semantic nor geometric labels in training and offers improved model flexibility and generalizability. Trajectory-regularized SSL further enhances geometric representations by incorporating an unsupervised geometry-trajectory-regularization loss.

Method	**semantic label**	**pose label**	**traj. reg.**
Fully-Supervised [40, 58]	✓	✓	✗
Geometry-Supervised [18, 26]	✗	✓	✗
Invariant SSL (baseline) [4, 10]	✗	✗	✗
Trajectory-Regularized SSL (ours)	✗	✗	✓

Supplementary Material

In this supplementary material, we provide details omitted in the main text including:

- Section A: Comparison with related methods and datasets from previous work;
- Section B: Implementation and results of the mid-layer representation compression, where we compress representations with minimal performance drop;
- Section C: Empirical study on the similarity between mid-layer features and patch embedding;
- Section D: Ablation study on the trajectory loss, non-equidistance pose of image triplets and the backbone architecture;
- Section E: Additional results on a large-scale dataset Objaverse [19].

A Method and Dataset Comparison

We compare with related works in Table 3. For dataset, we propose a benchmark with a dataset generation/rendering configuration that **1)** adheres to the self-supervised learning (SSL) setting where neither semantic nor geometric labels are used for training; **2)** allows evaluation on out-of-domain data with the introduction of the relative pose. We demonstrate the configuration on the ShapeNet dataset [8] as an example. There exist similar datasets derived from ShapeNet, such as 3DIEBench [26] and 3DIdent [59]. Although such datasets are designed for or suitable for benchmarking SSL geometric representations, we still provide comparisons in Table 4 given they are also derived from ShapeNet.

Table 4: Comparison with other datasets consisting of rendered images of objects from ShapeNet [8]. Our dataset **1)** does not use pose labels for training and adheres to SSL geometric representation evaluation setting; **2)** enables evaluation on out-of-domain data; **2)** has complete and even pose coverage for rendered images.

	Our dataset	**3DIEBench**	**3DIdent**
Out-of-domain evaluation	Yes	No	No
Pose coverage	$(-\pi, \pi)$	$(-\pi/2, \pi/2)$	$(-\pi/2, \pi/2)$
Pose sampling method	even	uneven	uneven
Numer of images	1.5M	2.5M	275k

B Compressing Mid-Layer Representations

Table 5: Mid-layer representations have higher pose estimation accuracies but lower efficiency due to high dimensionality. We show they can be compressed to lower dimensions with minimal performance drop for absolute pose estimation. For relative pose estimation, compressed features have a larger gap (4-5%) but outperform representations from the feature layer.

embedding	**# dim**	**abs. pose acc. (%)**	**rel. pose acc. (%)**
conv3	16,384	92.5	87.8
compressed conv3	512	91.4 (↓1.1)	82.4 (↓5.4)
conv4	8,192	91.9	85.2
compressed conv4	512	90.8 (↓1.1)	81.2 (↓4.0)
feature	512	87.8	77.5

Motivations and Methods. While mid-layer representations in networks like ResNet18 offer improved pose estimation accuracy, their large dimensions lead to inefficiencies. For instance, the "conv3" layer's dimension is twice that of "conv4" and 32 times larger than the pooled "feature" layer, resulting in inefficiency due to high dimensionality. To address this, we propose compressing mid-layer representations to lower dimensions using projection heads with multi-layer perceptrons. As depicted in Fig.3, we denote the "conv3" layer representation as $\mathbf{z}^3$ and the "conv4" layer representation as $\mathbf{z}^4$. We then use a projection head g_ϕ to reduce the dimensionality of these representations: for "conv3", $\mathbf{y}^3 = g_\phi^3(\mathbf{z}^3)$; and similarly for "conv4", $\mathbf{y}^4 = g_\phi^4(\mathbf{z}^4)$. More details are available in the supplementary.

Then the trajectory loss $\mathcal{L}_{\text{traj}}$ (Eqn.3) can be adapted for compressed feature y, e.g., when using "conv3" as the final representation, we can use the following trajectory loss:

$$\mathcal{L}_{\text{traj}}^{\text{conv3}}(\mathbf{y_L^3}, \mathbf{y_C^3}, \mathbf{y_R^3}) = \mathcal{L}_{\text{traj}}(g_\phi^3(\mathbf{z_L^3}), g_\phi^3(\mathbf{z_C^3}), g_\phi^3(\mathbf{z_R^3})) \tag{5}$$

Results. For fair comparison, we make the compressed mid-layer representation y has dimension of 512, the same as the dimension of feature-layer z. Our findings in Table 5 demonstrate that mid-layer features can be effectively condensed 32x into smaller dimensions as "feature"-layer with only a slight reduction in performance regarding absolute pose estimation (1%). In the case of relative pose estimation, while there is a more noticeable difference in performance (4%-5%) with compressed features, they still outperform the representations derived from the feature layer.

Implementation Details. For clarity, we provide details on compressing mid-layer representations of SimCLR [10] (Fig.9). For the semantic loss and downstream semantic classification, we always follow the baseline setting and

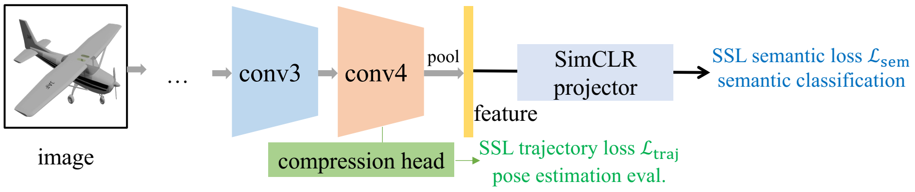


Fig. 9: We compress mid-layer representation from "conv4" layer, taking SimCLR [10] as an example. For the semantic loss, we follow SimCLR's setting and add the loss after SimCLR projector. For the pose loss, we use an MLP-based head to compress mid-layer features and the compressed feature to classify pose. Trajectory loss is put after the compression head.

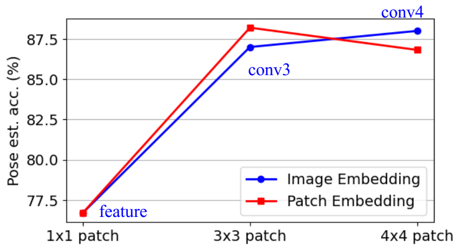


Fig. 10: Mid-layer representations improve SSL geometric representation quality, which could be partly attributed to the similarity to the patch embedding. Empirically, a similar trend of pose estimation accuracy gain was observed with patch embedding. The metric is relative pose estimation accuracy on in-domain data.

make no changes. We take SimCLR as an example. For pose estimation, we use an MLP-based head to compress mid-layer features and the compressed feature to classify pose. Trajectory is also put post-compression-head.

C Mid-Layer Features and Patch Embedding

As mentioned earlier, the improved SSL geometric representation quality by mid-layer representations could be partly attributed to the similarity to the patch embedding. Empirically, for the VICReg [4] baseline, we partition the input image to $m \times m$ patches ($m = 1, 3, 4$ in our experiment). As in Fig.10, using patch embedding has a similar effect as mid-layer representation and also improves the pose estimation accuracy.

D Ablation Study

Our examination focuses on VICReg with proposed trajectory regularization, using relative pose estimation as the task and the feature layer for evaluation.

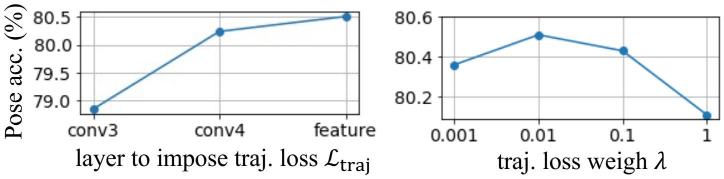


Fig. 11: Hyperparameter analysis on the trajectory-regularized VICReg, which is evaluated for relative pose estimation with representation being the feature-layer z. **Left:** While fixing the feature layer for the downstream task of pose estimation, we change different layers to impose the trajectory loss $\mathcal{L}_{\text{traj}}$. Feature-layer gives the best performance, although the difference is less than 2%. **Right:** The highest performance is achieved at trajectory loss weight $\lambda = 0.01$, though the method is not very sensitive to λ.

Layer for Trajectory Loss. In Fig.11**U**, we vary the layer utilized for the trajectory loss $\mathcal{L}_{\text{traj}}$ during training. Note that this is different from the setting in other experiments where trajectory loss is always constrained on feature z during training, and we change the layer as the representation for evaluation. The influence is $< 2\%$ for different layers.

Trajectory Loss Weight. In Fig.11**L**, the method exhibits a low sensitivity to changes in λ.

Non-Equidistant Poses. Our method works when the adjacent views in the trajectory loss are sampled from smooth trajectories, where the speed varies gradually. We show this with an empirical experiment in Table 6. Adjacent views exhibit non-equidistant poses during training: we randomly sample cubic Bézier curves with the starting pose p_L and ending pose p_R, where the angle between p_L, p_R is $(5°, 20°)$. The middle pose p_C is randomly sampled from the curve to simulate the speed variation. Non-equidistant pose trajectory regularization also gives 4% gain.

Different Backbones. We study if the performance gain of mid-layer representations generalizes to other network/backbone architectures. For VICReg [4] with trajectory loss, on ResNet50 backbone we also observe a similar trend of improvement with mid-level features as the ResNet18 backbone (Table 7).

Table 6: We render adjacent views that exhibit non-equidistant poses. Similar to equidistant poses, the trajectory loss with non-equidistant poses also gives 4% gain for relative pose estimation.

	Rel. pose acc(%)
VICReg	76.7
VICReg+equidistant traj.	80.5
VICReg+non-equidistant traj.	80.3

Table 7: For VICReg [4] with the proposed trajectory loss, we use different backbones and also observe performance gains of relative pose estimation accuracy with mid-layer representations.

Rel. pose acc(%)	feature	conv4	conv3
Res18	80.5	88.3	89.4
Res50	82.6	90.1	91.0

Table 8: Our trajectory regularization improves 1.3% relative pose estimation accuracy; with feature layer, ours has a 3.3% gain

Objaverse acc.	**conv4**		**feat**	
method	**VICReg**	**VICReg+traj.**	**VICReg**	**VICReg+traj.**
airplane	86.4	87.0(↑0.6)	77.9	81.9(↑4.0)
bench	90.3	92.1(↑1.8)	85.0	88.6(↑3.6)
car	91.0	91.9(↑0.9)	87.3	90.2(↑2.9)
chair	88.7	90.6(↑1.9)	83.2	87.3(↑4.1)
coffee table	88.6	90.0(↑1.4)	82.0	84.7(↑2.7)
gun	81.4	82.8(↑1.4)	70.6	73.2(↑2.6)
avg	87.8	89.1(↑1.3)	81.0	84.3(↑3.3)

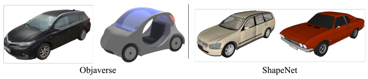


Fig. 12: Objaverse (left) has higher diversity than ShapeNet (right).

E Objaverse Results

We consider a 3D dataset with more diversity, Objaverse [19], with visual comparisons in Fig.12. We carry out the experiment on a subset of Objaverse [19], and the improvement is universal on every category. The semantic categories used in this experiment: airplane, bench, car, chair, coffee table and gun. Results show that the proposed trajectory regularization is effective and using mid-layer representation helps: with conv4 layer, our trajectory regularization improves 1.3% relative pose estimation accuracy; with feature layer, ours has a 3.3% gain (Table 8). The full-scale Objaverse experiment with comprehensive comparison will be included in the revision.