Supplementary Material

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

- Section A: Comparison with similar 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 Dataset Comparison

Our benchmark proposes 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

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

2 Wang et al.

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.

$\mathbf{embedding}$	$ \# \operatorname{dim}$	abs. pose acc. $(\%)$	rel. pose acc. (%)
conv3	16,384	92.5	87.8
compressed conv3	512	$91.4 (\downarrow 1.1)$	$82.4 (\downarrow 5.4)$
conv4	8,192	91.9	85.2
compressed conv4 $$	512	$90.8~(\downarrow 1.1)$	81.2 (↓4.0)
feature	512	87.8	77.5

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:

$$\mathcal{L}_{\text{traj}}^{\text{conv3}}(\mathbf{y}_{\mathbf{L}}^{\mathbf{3}}, \mathbf{y}_{\mathbf{C}}^{\mathbf{3}}, \mathbf{y}_{\mathbf{R}}^{\mathbf{3}}) = \mathcal{L}_{\text{traj}}(g_{\phi}^{3}(\mathbf{z}_{\mathbf{L}}^{\mathbf{3}}), g_{\phi}^{3}(\mathbf{z}_{\mathbf{C}}^{\mathbf{3}}), g_{\phi}^{3}(\mathbf{z}_{\mathbf{R}}^{\mathbf{3}}))$$
(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 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



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.

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.

Layer for Trajectory Loss. In Fig.11U, we vary the layer utilized for the trajectory loss \mathcal{L}_{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.11L, 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

Pose-Aware SSL

4 Wang et al.



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}_{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 .

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. **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(%)				
VICReg	76.7	Rel. pose $acc(\%)$	feature	conv4	conv3
VICReg+equidistant traj.	80.5	Res18	80.5	88.3	89.4
VICReg+non-equidistant traj.	80.3	Res50	82.6	90.1	91.0

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

Table 8: Our trajectory regularization improves1.3% relative pose estimation accuracy; with featurelayer, ours has a 3.3% gain

Objaverse ac	c.	conv4		feat
method	VICR	$\mathbf{eg} \mathbf{VICReg} + \mathbf{t} $	raj. VICR	${ m eg} { m VICReg+tra} $
airplane	86.4	87.0(<u></u>	77.9	81.9(†4.0)
bench	90.3	$92.1(\uparrow 1.8)$	85.0	88.6(†3.6)
car	91.0	91.9(^{10.9})	87.3	90.2(¹ 2.9)
chair	88.7	90.6(11.9)	83.2	87.3(14.1)
coffee table	88.6	90.0(† 1.4)	82.0	84.7(12.7)
gun	81.4	82.8(1.4)	70.6	73.2(†2.6)
avg	87.8	89.1(11.3)	81.0	84.3(†3.3)



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

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.