EquiCaps: Predictor-Free Pose-Aware Pre-Trained Capsule Networks

Supplementary Material

9. Implementation details

9.1. Reproducibility

All pre-training experiments employed three NVIDIA A100 80 GB GPUs and took approximately 24 hours. The code, weights, dataset, and dataset generation scripts are publicly released at http://github.com/AberdeenML/EquiCaps.

9.2. Pre-training

We adopt the experimental setup from [24]. All methods use ResNet-18 [30] as the base encoder network. For the compared methods except CapsIE, the projector is a three-layer MLP. For capsule-based methods, the projector consists of 32 capsules. Training lasts 2000 epochs with a batch size of 1024 to ensure convergence. The Adam optimiser [37] is employed with a learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 10^{-6}. We mention the hyperparameters of each method below.

Supervised ResNet-18 The training and evaluation protocols are identical to those of the self-supervised setups.

VICReg The projector is configured with intermediate dimensions of 2048-2048-2048, with loss weights $\lambda_{\text{inv}} = \lambda_{\text{V}} = 10$, and $\lambda_C = 1$.

SimCLR, SEN The projector is configured with intermediate dimensions of 2048-2048-2048, and the temperature parameter of the loss is set to 1.

SimCLR + AugSelf The projector is configured with intermediate dimensions of 2048-2048-2048, and the temperature parameter of the loss is set to 1. The parameter prediction head is configured with a MLP with intermediate dimensions 1024-1024-4. The two losses, SimCLR and parameter prediction, are assigned equal weight.

EquiMod In alignment with the original protocol, the projector is configured with intermediate dimensions of 1024-1024-128. The temperature parameter of the loss is set to 0.1. The two losses, invariance and equivariance, are assigned equal weight.

SIE Aligning with the original protocol, both the invariant and equivariant projectors are configured with intermediate dimensions 1024-1024-1024. The loss weights are $\lambda_{\text{inv}} = \lambda_V = 10$, $\lambda_{\text{equi}} = 4.5$, and $\lambda_C = 1$.

CapsIE, EquiCaps The projector is configured with 32 capsules. The loss weights are $\lambda_{\text{inv}} = 0.1$, $\lambda_{\text{equi}} = 5$, $\lambda_V = 10$, and $\lambda_C = 1$.

9.3. Evaluation

Semantic classification A linear classification layer is trained on the frozen representations. The Adam optimiser is used with a learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and no weight decay. Training is carried out for 300 epochs with a batch size of 256, using the cross-entropy loss function. Performance is evaluated on the validation set comprising objects excluded from the training set.

Rotation prediction A three-layer MLP with intermediate dimensions 1024-1024-4 and intermediate ReLU activations is trained on the frozen representations. The inputs are concatenated pairs of representations from two distinct views of an object. The MLP is trained to regress the rotation between these views. The Adam optimiser is used with a learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and no weight decay. Training is conducted for 300 epochs with a batch size of 256, using the mean-squared error loss function. Performance is evaluated on the validation set using R^2, which contains objects excluded from the training set, with the object rotation range the same across both sets.

Translation prediction The same methodology as for rotation prediction is followed. The only modification is that the output dimension of the MLP is three, corresponding to the elements of the translation vector, since is trained to regress the translation between the selected views.

Colour prediction A linear layer is trained on top of frozen representations to regress the floor and spot hue. The inputs are concatenated pairs of representations from two distinct views of an object. The Adam optimiser is used with a learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and no weight decay. Training is conducted for 50 epochs with a batch size of 256, using the mean-squared error loss function. Performance is evaluated on the validation set using R^2.

10. Sensitivity analysis & ablation studies

10.1. Convergence speed

We examine convergence speed on 3DIEBench and 3DIEBench-T. As shown in Tab. A.5 for rotation prediction on 3DIEBench, our method converges fastest, reaching an R^2 of 0.75 by 100 epochs and stabilising at 0.78 by 500 epochs, significantly ahead of the other methods and more than double the early-stage performance of SIE and CapsIE. As shown in Tab. A.6, a similar trend emerges in classification, where our method achieves the highest accuracy at 100 epochs—surpassing even the purely invariant methods. While longer training further improves our classification performance, our approach gains less from 500 to 2000 epochs compared to the rest methods.

Convergence-speed results on 3DIEBench-T for rotation prediction and classification appear in Tab. A.7 and Tab. A.8, respectively. As with the rotation prediction task on 3DIEBench, our method still converges the fastest, though its relative convergence speed between 100 and 500 epochs is slower. In classification, other methods exhibit comparable performance. We attribute this to the added complexity of 3DIEBench-T as learning more complex geometric transformations demands more epochs for EquiCaps to encode all transformations.

Table A.5. Impact of training duration on 3DIEBench rotation prediction performance on learned representations.

	Rotation (R^2)				
Method	100 ep.	500 ep.	1000 ep.	1500 ep.	2000 ep.
VICReg	0.28	0.46	0.46	0.46	0.45
SimCLR	0.42	0.48	0.49	0.51	0.52
SimCLR + AugSelf	0.50	0.71	0.74	0.75	0.75
SEN	0.39	0.48	0.49	0.50	0.51
EquiMod	0.40	0.48	0.49	0.49	0.50
SIE	0.29	0.68	0.72	0.73	0.73
CapsIE	0.31	0.68	0.75	0.75	0.74
EquiCaps	0.75	0.78	0.78	0.78	0.78

Table A.6. Impact of training duration on 3DIEBench classification performance on learned representations.

	Classification (Top-1)				
Method	100 ep.	500 ep.	1000 ep.	1500 ep.	2000 ep.
VICReg	49.12	79.40	83.12	84.43	84.28
SimCLR	72.58	84.00	85.87	86.61	86.73
SimCLR + AugSelf	73.50	84.57	86.51	87.08	87.44
SEN	67.24	82.36	85.29	86.45	86.99
EquiMod	73.19	84.89	86.36	87.00	87.39
SIE	51.49	77.59	81.05	82.12	82.94
CapsIE	46.12	72.60	78.68	79.54	79.35
EquiCaps	75.44	81.10	82.36	82.82	83.24

Table A.7. Impact of training duration on 3DIEBench-T rotation performance on learned representations. Equivariant methods are pre-trained for both rotation and translation.

	Rotation (R^2)				
Method	100 ep.	500 ep.	1000 ep.	1500 ep.	2000 ep.
VICReg	0.25	0.24	0.25	0.36	0.39
SimCLR	0.37	0.44	0.44	0.45	0.44
SimCLR + AugSelf	0.41	0.63	0.67	0.68	0.69
SEN	0.34	0.46	0.46	0.46	0.46
EquiMod	0.38	0.46	0.46	0.45	0.46
SIE	0.26	0.44	0.47	0.48	0.48
CapsIE	0.25	0.49	0.62	0.62	0.62
EquiCaps	0.54	0.70	0.71	0.70	0.71

Table A.8. Impact of training duration on 3DIEBench-T classification performance on learned representations. Equivariant methods are pre-trained for both rotation and translation.

	Classification (Top-1)				
Method	100 ep.	500 ep.	1000 ep.	1500 ep.	2000 ep.
VICReg	31.50	19.46	31.80	66.60	74.71
SimCLR	67.56	78.32	79.52	79.87	80.08
SimCLR + AugSelf	66.02	78.80	80.04	80.77	81.04
SEN	59.64	77.11	79.17	79.66	80.23
EquiMod	69.02	79.31	80.38	80.85	80.89
SIE	44.97	67.72	73.92	75.58	75.91
CapsIE	37.22	65.14	74.13	75.63	76.31
EquiCaps	62.00	75.87	77.31	77.45	77.88

10.2. Number of capsules

We report in Tab. A.9 that increasing the number of capsules improves classification and rotation prediction on 3DIEBench. A similar trend appears for 3DIEBench-T, though geometric tasks depend on explicit optimisation. Specifically, from 32 to 64 capsules yields a slight decrease in translation when optimising only for rotation, suggesting that the additional capacity is primarily dedicated to the explicit rotation objective. Nonetheless, when we optimise for both objectives, translation performance increases substantially, confirming the effectiveness of the 4×4 capsule pose structure and our proposed matrix manipulation while classification, rotation and translation performance remain near supervised level. We also observe that as the number of capsules grows, the network tends to focus more on geometric tasks relative to colour prediction.

Table A.9. Impact of the number of capsules in the EquiCaps projector on 3DIEBench and 3DIEBench-T. †Denotes pre-trained for both rotation and translation. We evaluate invariance via classification and equivariance via rotation, translation, and colour prediction tasks.

	Classification (Top-1)			Rotation (R^2)			Translation (R^2)		Colour (R^2)		
No. of Capsules	3DIEBench	3DIEBench-T		3DIEBench	3DIEBench-T		3DIEBench-T		3DIEBench	3DIEBench-T	
16	81.89	73.86	74.13†	0.77	0.71	0.67†	0.57	0.56†	0.13	0.04	0.04†
32	83.24	76.91	77.88†	0.78	0.73	0.71†	0.60	0.61†	0.09	0.05	0.02†
64	83.66	77.96	78.80†	0.79	0.74	0.71†	0.53	0.64†	0.05	0.01	0.01†

Table A.10. Ablation study of different loss function components contributing to invariance in EquiCaps, evaluated on 3DIEBench downstream tasks. We evaluate learned representations on invariance (classification) and equivariance (rotation and colour prediction via R^2). The losses considered are $\mathcal{L}_{\text{inv}}$ (invariance), $\mathcal{L}_{\text{ME-MAX}}$ (mean entropy maximisation regularisation), $\mathcal{L}_{\text{equi}}$ (equivariance), and $\mathcal{L}_{\text{reg}}$ (variance-covariance regularisation). $\mathcal{L}_{\text{reg}}$ can be applied either to Z_{cat}, concatenating both activation and pose matrices, or to Z_{pose}, which is applied only to the pose matrices.

	Loss Functions				Classification (Top-1)	Rotation (R^2)	Colour (R^2)
Method	$\mathcal{L}_{\text{inv}}$	$\mathcal{L}_{\text{ME-MAX}}$	$\mathcal{L}_{\text{equi}}$	$\mathcal{L}_{\text{reg}}$			
EquiCaps OnlyEqui	-	-	✓	Z_{pose}	81.68	0.78	0.01
EquiCaps w/o $\mathcal{L}_{\text{ME-MAX}}$ & $\mathcal{L}_{\text{inv}}$	-	-	✓	Z_{cat}	82.40	0.78	0.04
EquiCaps OnlyEqui w/ $\mathcal{L}_{\text{ME-MAX}}$	-	✓	✓	Z_{pose}	81.61	0.78	0.06
EquiCaps w/o $\mathcal{L}_{\text{inv}}$	-	✓	✓	Z_{cat}	81.43	0.78	0.06
EquiCaps	✓	✓	✓	Z_{cat}	83.24	0.78	0.09

10.3. Invariance loss function: components

We examine how different invariance-related loss function components influence the performance of EquiCaps on downstream tasks using the 3DIEBench. Specifically, we evaluate the impact on classification (invariance), rotation and colour prediction (equivariance), on learned representations. We investigate the following configurations:

(a) $\mathcal{L}_{\text{equi}} + \mathcal{L}_{\text{reg}}(Z_{\text{pose}})$: Equivariance combined with variance-covariance regularisation applied only to the pose matrices.

(b) $\mathcal{L}_{\text{equi}} + \mathcal{L}_{\text{reg}}(Z_{\text{cat}})$: Equivariance combined with variance-covariance regularisation applied on the concatenated activation vectors and pose matrices.

(c) $\mathcal{L}_{\text{equi}} + \mathcal{L}_{\text{reg}}(Z_{\text{pose}}) + \mathcal{L}_{\text{ME-MAX}}$: As in (a), but combined with mean-entropy maximisation regularisation ($\mathcal{L}_{\text{ME-MAX}}$).

(d) $\mathcal{L}_{\text{equi}} + \mathcal{L}_{\text{reg}}(Z_{\text{cat}}) + \mathcal{L}_{\text{ME-MAX}}$: As in (b), but combined with $\mathcal{L}_{\text{ME-MAX}}$.

(e) Our EquiCaps method as defined in Eq. (8) that combines invariance, equivariance, variance-covariance regularisation applied on the concatenated activation vectors and pose matrices, and mean-entropy maximisation regularisation.

We observe in Tab. A.10 that including all of our invariance-related losses $\mathcal{L}_{\text{inv}}$, $\mathcal{L}_{\text{ME-MAX}}$, $\mathcal{L}_{\text{reg}}(Z_{\text{cat}})$, combined with $\mathcal{L}_{\text{equi}}$ improve classification without compromising rotation performance, and we maintain almost perfect invariance to colour hue transformations. As expected, the mean-entropy maximisation $\mathcal{L}_{\text{ME-MAX}}$ alone does not suffice to boost classification in the absence of our invariance loss, regardless of whether the variance-covariance regularisation is applied to the pose matrices or to the concatenated embeddings. This finding suggests that $\mathcal{L}_{\text{ME-MAX}}$ most effectively distributes activations when combined with the invariance loss. Similarly, whether we apply the variance-covariance regularisation solely to the pose matrices or to the concatenated embeddings, including the invariance loss and $\mathcal{L}_{\text{ME-MAX}}$ still yields further gains in classification. We also find that in every ablation setting, rotation performance remains stable and on par with the supervised baseline. Nonetheless, our proposed method further enhances semantic representation while retaining almost perfect invariance to colour.

Table A.11. Evaluation on a subset of Objaverse-LVIS using a ResNet-18 backbone. Representations are evaluated on an invariant task (classification) and an equivariant task (rotation prediction). Each model is evaluated five times with different random seeds. We report in bold the best performance across all methods.

	Pre-training (Frozen Backbone)		Transfer Learning (Fine-tuning)	
Method	Classification (Top-1)	Rotation (R^2)	Classification (Top-1)	Rotation (R^2)
VICReg	80.43±1.11	29.06±0.79	90.22±0.70	62.06±1.17
SimCLR	83.44±0.88	28.80±0.34	91.08±0.72	63.36±1.10
AugSelf	83.87±0.38	29.80±1.49	90.75±0.70	63.69±0.88
SEN	82.90±1.17	29.51±1.14	90.86±0.38	63.61±1.52
EquiMod	**83.76**±0.45	29.58±1.25	89.89±0.24	62.84±1.09
SIE	75.27±1.26	28.96±0.49	89.78±0.66	61.85±0.70
CapsIE	72.58±0.85	43.32±1.39	90.75±0.45	63.60±0.95
EquiCaps	78.82±0.61	**49.46**±0.91	**92.80**±0.61	**65.14**±1.12

Table A.12. Transfer learning via DETR fine-tuning on MOVi-E. We report in bold the best performance.

Method	Classification (Top-1)	Rotation (R^2)	mAP	mAP_{50}	mAP_{75}
SIE	73.7	0.20	26.47	41.83	28.26
CapsIE	73.3	0.21	27.03	41.97	29.84
EquiCaps	**75.2**	**0.24**	**30.91**	**48.74**	**33.58**

11. Additional quantitative results

11.1. Objaverse results

Unlike ShapeNet-derived datasets [9] such as 3DIEBench [24] and our 3DIEBench-T, Objaverse [16] contains a considerably wider variety of objects, many of which are real-world scans. To further validate EquiCaps on a different randomly rotated multi-view dataset, we use a subset of Objaverse-LVIS [16] with six classes (airplane, bench, car automobile, chair, coffee table, and gun), following the class selection of [54]. We evaluate classification and rotation prediction using two approaches. First, we perform transfer learning by fine-tuning the whole network pre-trained on 3DIEBench. Second, we pre-train each model from scratch, and we train only the task-specific heads. All remaining experimental settings are identical to those described in supplementary Sec. 9.

As the fine-tuning results in Tab. A.11 show, EquiCaps generalises best on both tasks, outperforming even the invariant methods. With the encoder frozen, the classification results for all methods are similar to their 3DIEBench results. For rotation prediction, EquiCaps maintains the best performance, while all methods except the capsule-based CapsIE show a similar drop in performance. We attribute this decline to the lack of an explicit pose mechanism in non-capsule architectures, which makes pose learning more difficult on small datasets, whereas CapsNets have shown improved performance on small datasets [15].

11.2. MOVi-E results

To further evaluate the performance in more realistic and challenging setting we explore the Multi-Object Video (MOVi-E) dataset [26], which includes synthetic generation of multiple objects, including occlusions, and realistic backgrounds. The dataset consists of random scenes with up to 17 distinct objects placed in realistic rendered environments. Each scene is rendered with a 2 second rigid body simulation with multiple objects falling. MOVi-E uses a linear camera movement, but in our setting we do not process the video sequence, but instead sample frames and process each independently. For full details on the dataset, we refer the reader to the original work [26] and the dataset repository[1].

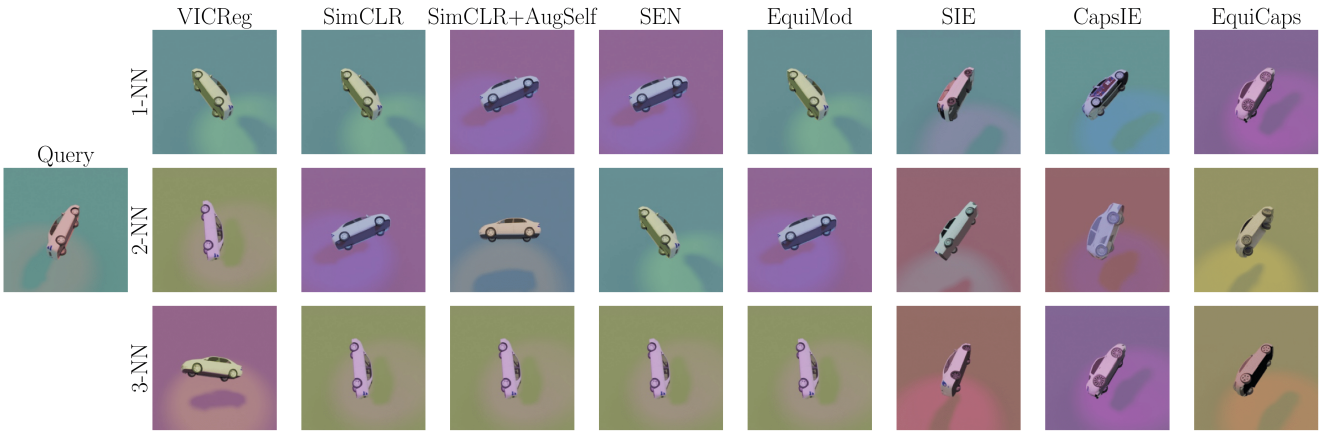
For this alternative dataset, our task is to both detect each of the objects via a bounding box, and subsequently determine the object type (classification) and its pose in relation to the frame of the camera. To accommodate object detection, we employ the DETR [5] architecture. The ResNet-50 backbone is directly taken from pre-training on 3DIEBench and weights transferred to the backbone of the DETR architecture. We then freeze the backbone, and fine-tune the transformer encoder, decoder, and prediction heads via the MOVi-E dataset. We modify DETR by adding a MLP predictor to regress the rotation quaternions for each object. This quaternion predictor takes the same form as the bounding box predictor, with a three-layer MLP with 256 hidden dimensions. The quaternion regression is optimised via minimising the mean-squared error and this term is simply added to the overall objective, which is a weighted sum of individual losses, details of which are presented in [5].

We train the transformer encoder, decoder, and prediction heads for 200 epochs, with a batch size of 64, and using a learning rate of 0.0001 reduced by a factor of 10 at epoch 100. We set the weighting of the quaternion loss to 2 and Generalized Intersection over Union to 3, leaving all other settings to default as defined in the original work.

The results of the classification, object detection, and rotation regression are given in Tab. A.12. We observe that our approach achieves improved performance across all evaluation metrics compared to SIE and CapSIE, which are the next best equivariant methods. This demonstrates improved generalisation and robustness of the learned representations to operate in more complex settings. Although EquiCaps's performance is superior in our experiments, the overall performance is still limited given the inherent complexity of the dataset and that the ResNet-50 backbone has been trained on the single-object setting. Future work will explore this setting in more detail and introduce adaptations to process video sequences.

[1] https://github.com/google-research/kubric/blob/main/challenges/movi.

Figure A.6. Nearest-neighbour representation retrieval on 3DIEBench validation set directly after pre-training. The query image (left) is compared against each method's learned representations to find its top three nearest neighbours (in rows: 1-NN, 2-NN, 3-NN).



12. Additional qualitative results

12.1. Nearest-neighbour representations on 3DIEBench

We replicate the retrieval of nearest representations from Fig. 4 performed on the 3DIEBench-T dataset, on 3DIEBench and show the results in Fig. A.6. We observe that our method, CapsIE, and SIE consistently retrieve nearest neighbours in similar poses as with the query, consistent with their high quantitative rotation prediction results. For the remaining methods, we observe that the learned invariance dominates, yielding mostly a range of object poses among the retrieved samples. However, aside from VICReg which showed the highest level of invariance in the quantitative results, these methods also retrieve some nearest neighbours in similar poses—albeit less accurately—demonstrating that partial rotation-related information remains at the representation stage. Overall, these findings are consistent with our quantitative results.

12.2. Equivariance via pose manipulation

In addition to the qualitative results shown in Fig. 5, we further illustrate our method's equivariant properties by performing the inverse of the previous experiment. Specifically, we rotate an object around the z-axis within the range $[0, \frac{\pi}{2}]$ in 5° increments, and feed each of these generated and original (source) pose into our projector to obtain its pose embeddings. Next, instead of transforming each embedding by the inverse rotation, we multiply the source pose embedding by the corresponding transformation matrix as shown in Fig. A.7. For each transformed embedding we retrieve the nearest neighbour among all embeddings. Observing how each nearest neighbour closely changes according to the applied latent transformation further highlights our methods' equivariant properties and its capability to preserve and manipulate pose information directly in the latent space.

Figure A.7. Illustration of equivariant capsule-based pose manipulation. We observe that our pose embeddings change predictably based on the applied transformation.

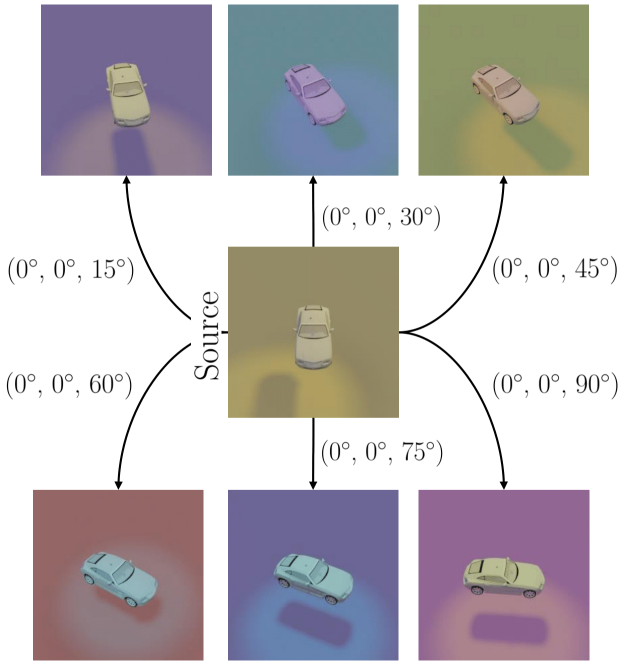


Table A.13. Computational cost during pre-training on 3DIEBench using three NVIDIA A100 80GB GPUs with a ResNet-18 backbone and batch size 1024. Equivariant methods are trained to be solely rotation-equivariant.

Method	No. of Parameters (M)	Iteration Time (s)	GFLOPs
VICReg	20.6	0.64	12.3
SimCLR	20.6	0.65	12.3
SimCLR + AugSelf	22.7	0.74	12.3
SEN	24.8	0.65	12.3
EquiMod	14.6	0.65	12.3
SIE	20.1	0.65	12.3
CapsIE	32.6	1.16	12.9
EquiCaps	18.6	0.90	12.7

13. Computational cost

In Tab. A.13 we compare, for each method, the number of parameters, iteration time, and GFLOPs during pre-training on the 3DIEBench dataset. The iteration time is computed over 1000 iterations after 1000 warmup iterations. The rest of the settings are identical to those described in Sec. 9.2.

We observe that adding the CapsNet head in EquiCaps increases computation moderately compared to the non-capsule-based approaches, due to use of the non-iterative self-routing algorithm [29]. This is a reasonable trade-off, as EquiCaps achieves state-of-the-art SO(3) and SE(3) equivariance. Notably, EquiCaps is more computationally efficient than previous capsule-based approaches because it enforces equivariance directly in the pose matrices, omitting the need for the intermediate predictions required by the separate predictor module.

14. 3DIEBench-T dataset generation

14.1. Generation protocol

3DIEBench-T extends the original settings of the 3DIEBench dataset by incorporating object translations. The dataset consists of 52,472 object instances across 55 classes from ShapeNetCoreV2 which are originally sourced from 3D Warehouse.

For each instance, 50 random views are drawn from a uniform distribution over the parameter ranges listed in Tab. A.14, producing images at a resolution of 256×256 pixels. For each view, the object is first translated by t and then rotated by R, thus, the object's final base translation is Rt. We also store the transformation parameters as latent information alongside each image, facilitating equivariant tasks. We are following the original settings of 3DIEBench in which the rotation ranges are constrained to make the task more controllable and the lighting angle is adjusted to ensure that shadows do not provide a trivial shortcut for the model. Translation ranges are restricted so that objects do not move outside the camera's view. Following the 3DIEBench split protocol, 80% of the objects are used for training, while the remaining 20%, unseen during training but sampled from the same transformation distribution, form the validation set. Generating the entire dataset requires approximately 44 hours on 12 NVIDIA A100 80GB GPUs, though the process can run in a single script.

Table A.14. Parameter ranges for uniformly random object rotation, translation, and lighting in 3DIEBench-T. Tait–Bryan angles are used to define extrinsic object rotations, and the light's position is specified using spherical coordinates.

Parameter	Min. Value	Max. Value
Object rotation X	$-\pi/2$	$\pi/2$
Object rotation Y	$-\pi/2$	$\pi/2$
Object rotation Z	$-\pi/2$	$\pi/2$
Object translation X	-0.5	0.5
Object translation Y	-0.5	0.5
Object translation Z	-0.5	0.5
Floor hue	0	1
Light hue	0	1
Light θ	0	$\pi/4$
Light ϕ	0	2π

14.2. Supplementary image samples

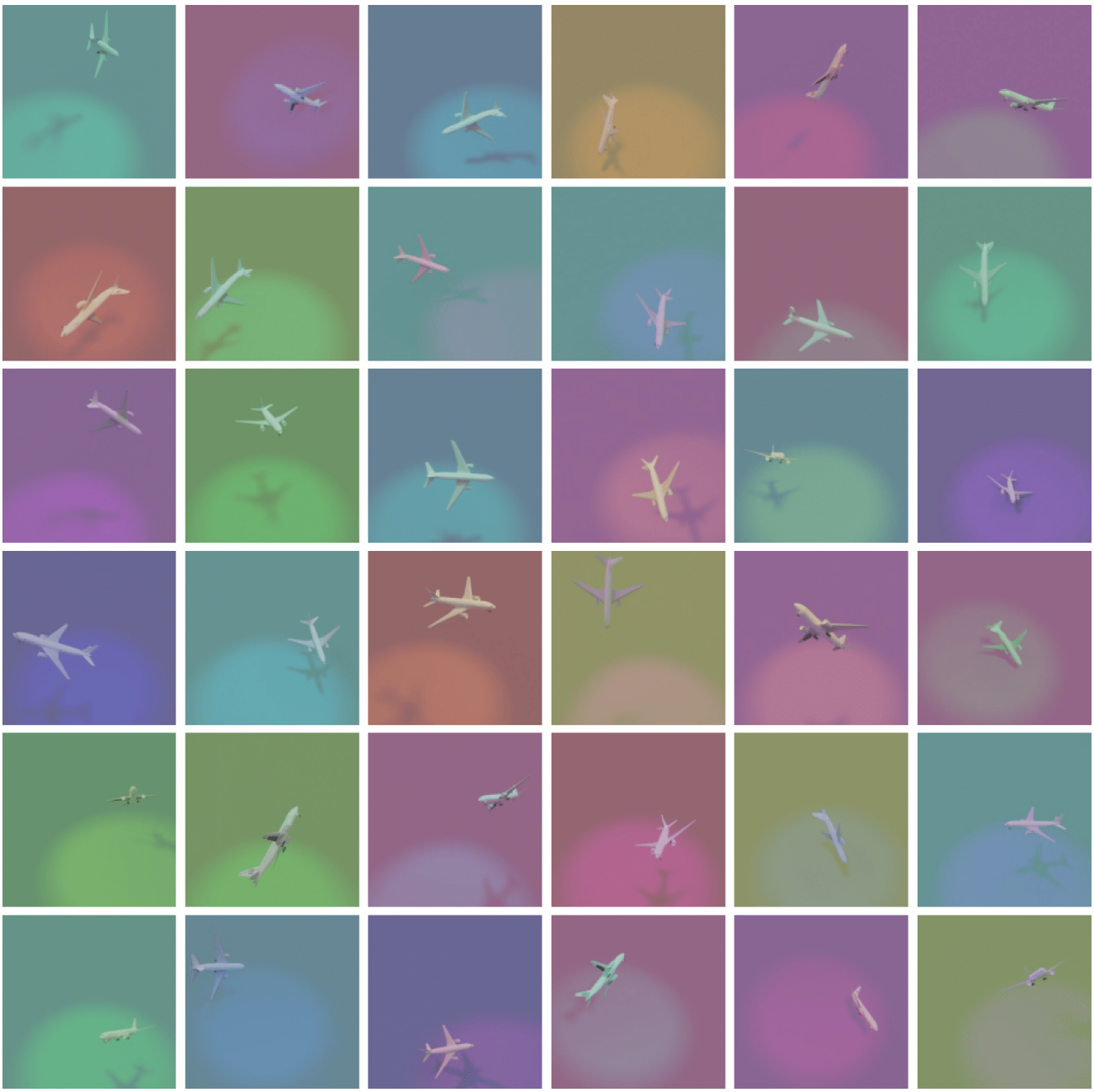


Figure A.8. Samples of an object instance from the 3DIEBench-T dataset.

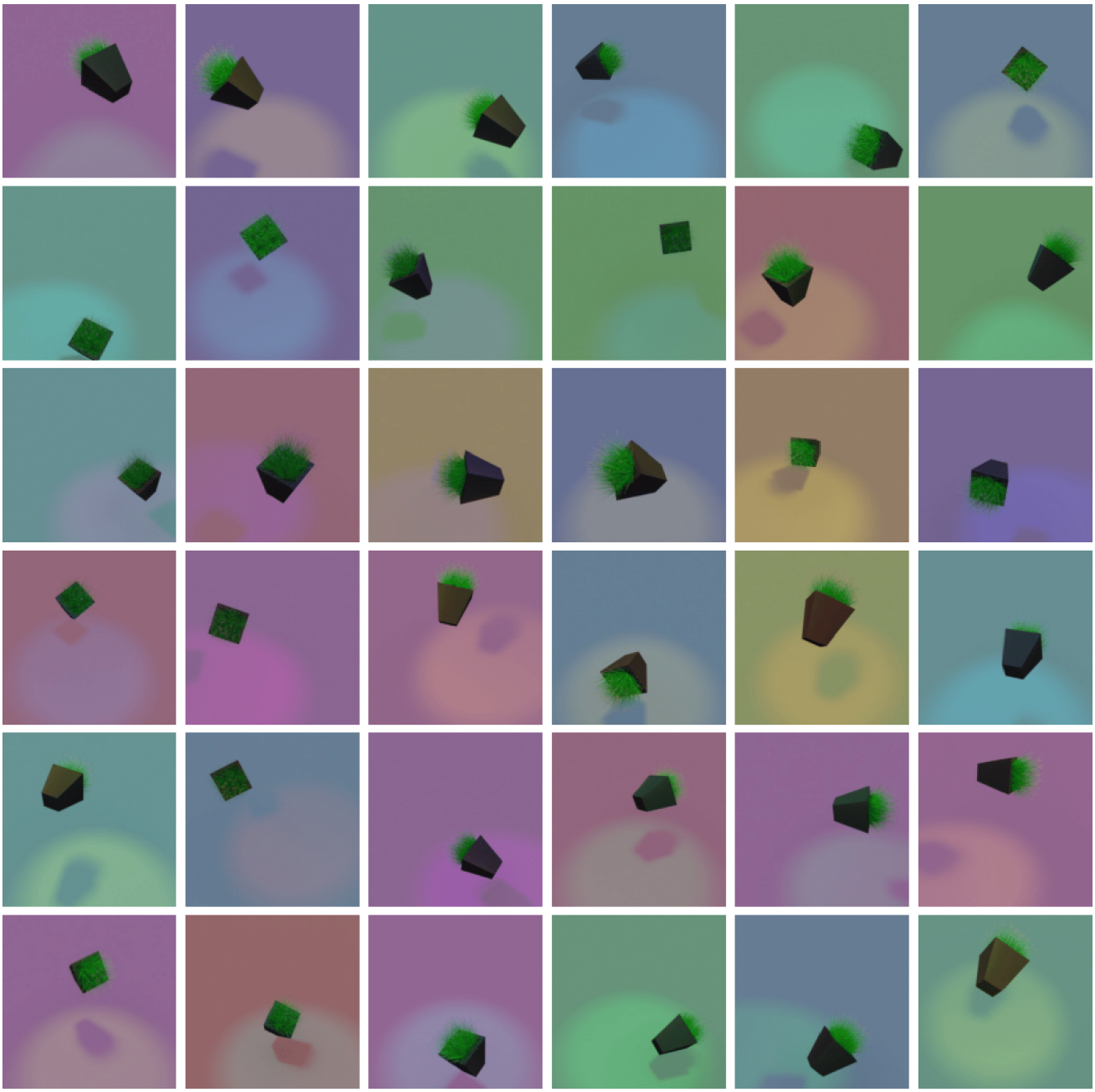


Figure A.9. Samples of an object instance from the 3DIEBench-T dataset.

Figure A.10. Samples of object instances from the 3DIEBench-T dataset.