Probing the Mid-level Vision Capabilities of Self-Supervised Learning

Supplementary Material

Sec. A provides an overview the self-supervised learning models (Tab. 5) included in our study. Sec. B details the evaluation metrics and presents the quantitative results (Tab. 6 - 11) for each mid-level vision task. Sec. D showcases qualitative visualizations (Fig. 8 - 9). Sec. C discusses the potential impact of the DPT head.

A. Self-supervised Learning Models

In our experiments, we select 22 SSL models from a wide range of categories based on two criteria: (1) coverage of the main approaches used for large-scale self-supervised training and (2) comparable model architecture and training data to allow fair comparisons. We primarily evaluate the publicly-available checkpoints pretrained on ImageNet1K [12] — the links to each checkpoint are included in Tab. 5. We briefly describe each SSL below.

Jigsaw. Noroozi and Favaro [45] introduced a self-supervised learning approach for model pretraining based on solving jigsaw puzzles as a pretext task. This method trains a network to predict the correct arrangement of shuffled image patches, where the image is divided into a 3x3 grid. At its core, this approach encourages the model to learn spatial relationships and understand object structure by generating consistent embeddings for the spatially rearranged patches of the same image. In our study, we used the publicly available ResNet-50 checkpoint trained on the ImageNet-1k [53] dataset.

Rotnet. Gidaris *et al*. [25] proposed a self-supervised approach for model pretraining using a rotation prediction task, known as RotNet. This method trains a network to classify the rotation angle (0°, 90°, 180°, or 270°) applied to an input image, encouraging the model to learn semantic features and spatial structure within the image. At its core, this approach leverages rotation as a proxy task, pushing the network to recognize objects and their orientations. In our work, we evaluate the ResNet-50 architecture trained on ImageNet-1k [53] using this pretext task and rely on the checkpoint released by the authors.

NPID. Wu *et al*. [66] introduced a non-parametric instance-level discrimination approach for unsupervised feature learning. This method trains a network to distinguish between individual instances by treating each image as its own unique class, employing a memory bank to store and update embeddings for all instances in the dataset. At its core, this approach promotes the model to learn discriminative features by maximizing the similarity between augmentations of the same instance and minimizing it across others. In our work, we evaluate the ResNet-50 architecture pre-trained on ImageNet-1k [53] using this instance discrimination task.

NPID++. Misra *et al*. [42] significantly improves upon the original implementation of NPID, achieving results that substantially outperform those reported in the original paper [66].

PIRL. Misra *et al*. [42] introduced Self-Supervised Learning of Pretext-Invariant Representations (PIRL), a method designed to learn representations that remain invariant across various pretext tasks. The approach applies contrastive learning, where the model is trained to produce similar embeddings for multiple augmentations of the same image while distinguishing between different images. At its core, PIRL combines instance discrimination with pretext invariance to capture both semantic and structural features. In our work, we evaluate the ResNet-50 architecture pre-trained on ImageNet using the PIRL framework.

ClusterFit. Yan *et al*. [67] proposed ClusterFit, a self-supervised learning approach that improves feature representations through clustering and re-training. This method begins by clustering embeddings of unlabeled images to capture the underlying data distribution, using these cluster assignments as pseudo-labels to retrain the model, thus distilling semantic information at the cluster level. At its core, ClusterFit follows a two-step process—clustering followed by supervised re-training—to develop robust and discriminative features. In our work, we evaluate the checkpoint using ResNet-50 architecture which is pre-trained on ImageNet.

SimCLR. Chen *et al*. [6] proposed SimCLR, a contrastive self-supervised learning framework designed to learn visual representations by maximizing agreement between different augmented views of the same image. The method applies a series of data augmentations, including random cropping, color distortion, and Gaussian blur, and uses a contrastive loss to bring embeddings of the same image instance closer together while pushing apart embeddings of different images. At its core, SimCLR leverages a simple yet effective contrastive objective, removing the need for specialized ar-

Table 5. **Self-Supervised Model Details.** This table provides details about each model, including the backbone architecture, the dataset used for training, and the source links to the checkpoints utilized in our experiments.

Model Name	Backbone	Dataset	Source Link
Jigsaw [45]	ResNet-50	ImageNet-1K	VISSL model zoo
RotNet [25]	ResNet-50	ImageNet-1K	VISSL model zoo
NPID [66]	ResNet-50	ImageNet-1K	VISSL model zoo
SeLa-v2 [4]	ResNet-50	ImageNet-1K	SwAV repository
NPID++ [42]	ResNet-50	ImageNet-1K	VISSL model zoo
PIRL [42]	ResNet-50	ImageNet-1K	VISSL model zoo
ClusterFit [67]	ResNet-50	ImageNet-1K	VISSL model zxwoo
DeepCluster-v2 [4]	ResNet-50	ImageNet-1K	SwAV repository
SwAV [4]	ResNet-50	ImageNet-1K	SwAV repository
SimCLR [6]	ResNet-50	ImageNet-1K	VISSL model zoo
MoCo v2 [8]	ResNet-50	ImageNet-1K	MoCo v2 repository
SimSiam [7]	ResNet-50	ImageNet-1K	MMSelfSup model zoo
BYOL [27]	ResNet-50	ImageNet-1K	Unofficial BYOL repo
Barlow Twins [69]	ResNet-50	ImageNet-1K	MMSelfSup model zoo
DenseCL [58]	ResNet-50	ImageNet-1K	DenseCL repository
DINO [5]	ResNet-50/ViT-B/16	ImageNet-1K	DINO repository
MoCo v3 [9]	ResNet-50/ViT-B/16	ImageNet-1K	MoCo v3 repository
iBOT [71]	ViT-B/16	ImageNet-1K	iBOT repository
MAE [30]	ViT-B/16	ImageNet-1K	MAE repository
MaskFeat [62]	ViT-B/16	ImageNet-1K	MMSelfSup model zoo

chitectures or memory banks. In our work, we evaluate the ResNet-50 architecture trained on ImageNet-1k [53].

SwAV. Caron *et al*. [4] introduced SwAV (Swapping Assignments between Views), a self-supervised learning approach that combines clustering with contrastive learning. Instead of directly contrasting augmented views, SwAV clusters the features of one view and assigns pseudo-labels, which are then used to predict the cluster assignments of another view. This method enables the model to learn representations without requiring negative samples or a memory bank. At its core, SwAV maximizes similarity between different augmentations by leveraging these swapped cluster assignments. In our work, we evaluate the ResNet-50 architecture trained on ImageNet 1k with SwAV.

SeLa-v2. SeLa [1] proposes an alternative approach to clustering-based self-supervised learning by formulating the clustering process as an optimization problem. It uses the Sinkhorn-Knopp algorithm to solve this optimization efficiently, ensuring that cluster assignments are balanced across the dataset. This avoids degenerate solutions where all data points are assigned to a single cluster. Caron *et al*. [4] re-implemented SeLa which improves upon the original SeLa by incorporating additional training improvements introduced in the self-supervised learning literature, such as stronger data augmentation, an MLP projection head, and temperature scaling for contrastive learning and yields better performance.

MoCo-v2. Chen *et al*. [8] proposed MoCo-v2, an improved version of the Momentum Contrast (MoCo) framework for self-supervised learning. MoCo-v2 enhances the original MoCo by incorporating stronger data augmentations (such as color distortion and Gaussian blur) and using an MLP projection head to further improve representation quality. Similar to its predecessor, MoCo-v2 employs a memory bank to maintain a large pool of negative samples and uses a momentum-updated encoder to produce stable representations. At its core, this approach refines instance discrimination with updated augmentations and architecture adjustments. In our work, we evaluate the ResNet-50 architecture trained on ImageNet using MoCo-v2.

SimSiam. Chen and He [7] proposed SimSiam, a self-supervised learning framework designed to simplify contrastive learning by removing the need for negative samples, momentum encoders, or memory banks. Instead, SimSiam trains a Siamese network with two branches, where one branch predicts the representation of the other. By using only a stop-gradient operation on one branch, SimSiam prevents the network from collapsing to trivial solutions, allowing it to learn meaningful representations from positive pairs alone. At its core, SimSiam is a simple and efficient method that demonstrates the feasibility of contrastive learning without negatives. In our work, we evaluate the ResNet-50 architecture trained on ImageNet 1k with SimSiam.

DenseCL. Wang *et al*. [58] introduced DenseCL, a self-supervised learning approach that extends contrastive learn-

ing to dense feature correspondences within images. Unlike traditional contrastive methods focused on global representations, DenseCL aims to learn pixel-level features by contrasting dense local regions between augmented views of the same image. This pixel-level contrastive objective encourages the model to learn spatially detailed representations, which benefit dense prediction tasks such as object detection and segmentation. At its core, DenseCL leverages fine-grained contrastive learning to produce more spatially aware features. In our work, we evaluate the ResNet-50 architecture trained on ImageNet 1k using DenseCL.

BYOL. Grill *et al*. [27] proposed BYOL, a self-supervised learning framework that learns visual representations without requiring negative samples. BYOL employs two neural networks: a "student" network and a "target" network. The student learns to predict the target's representation of an augmented view of the same image, and the target network is updated as an exponential moving average of the student. This setup enables the model to avoid trivial solutions by progressively refining representations through self-distillation. At its core, BYOL relies on bootstrap mechanisms and a momentum update to learn meaningful features without contrastive pairs. In our work, we evaluate the ResNet-50 architecture trained on ImageNet 1k using BYOL.

DeepCluster-v2. Caron *et al*. [3] introduced DeepCluster which uses k-means clustering on deep features to assign pseudo-labels to unlabeled data. These pseudo-labels are then used for training the network in an iterative process. However, DeepCluster suffers from the instability of cluster assignments between epochs, which requires reinitializing the classification layer repeatedly, disrupting the training of the convolutional network. Caron *et al*. [4] reimplement DeepCluster and address ealier issues by introducing explicit comparisons between features and cluster centroids instead of learning a classification layer for cluster assignments. This direct comparison increases the stability and performance of the training process. Additionally, DeepCluster-v2 incorporates modern self-supervised learning tricks and further enhances the method's performances.

Barlow Twins. Zbontar *et al*. [69] proposed Barlow Twins, a self-supervised learning approach designed to reduce redundancy in representations by decorrelating feature dimensions. The method uses a loss function that encourages the cross-correlation matrix between two identical networks' embeddings of augmented views to be as close to the identity matrix as possible, reducing redundancy across dimensions. This setup allows the model to learn diverse and informative features without the need for negative samples or memory banks. At its core, Barlow Twins promotes redundancy reduction, enhancing feature decorrelation. In our work, we evaluate the ResNet-50 architecture pre-trained on ImageNet 1k using Barlow Twins.

MoCo-v3. Chen *et al*. [9] proposed MoCo-v3, an extension of the Momentum Contrast framework tailored for Vision Transformers (ViTs) in self-supervised learning. MoCo-v3 adapts the momentum contrastive learning strategy to ViTs, introducing optimizations such as an MLP projection head and advanced data augmentations. Similar to previous versions, MoCo-v3 leverages a momentum-updated encoder to generate stable features and uses a queue-based memory bank to manage negative samples. At its core, this approach refines contrastive learning by combining MoCo's momentum mechanism with the ViT architecture. In our work, we evaluate the ViT-B/16 architecture trained on ImageNet using MoCo-v3 and employ the checkpoint released by the authors.

DINO. Caron *et al*. [5] proposed a self-distillation approach for model pretraining. The proposed approach trains a student network to generate features similar to a teacher network, where the teacher is an exponential moving average of the student network. At its core, this approach relies on instance discrimination as the model is trained to learn to generate similar embeddings for different crops of the same image instance. In our work, we evaluate the ViT-B/16 architecture trained on ImageNet-1k. We use the checkpoint released by the authors.

MAE. He *et al*. [30] showed that training vision transformers to reconstruct images based on randomly masked inputs is an effective pretraining task. Such models are trained with a large masking ratio; e.g., 75% of the input image patches are masked. In our experiments, we use the ViTB/16 and ViT-L/16 models trained on ImageNet-1k.

MaskFeat. Wei *et al*. [62] introduced MaskFeat, a self-supervised learning approach that learns visual representations by predicting masked visual tokens in videos. MaskFeat leverages a Vision Transformer (ViT) and operates by masking random patches in input video frames, then training the model to predict feature embeddings of these masked regions. This strategy encourages the model to capture rich semantic and spatial features, which generalize well across various downstream tasks. At its core, MaskFeat combines masked prediction with a ViT backbone, making it particularly effective for dense prediction tasks. In our work, we evaluate the ViT-B/16 architecture trained on ImageNet-1k using MaskFeat.

BEiT-v2. Peng *et al*. [48] proposed BEiT-v2, a self-supervised learning method that improves upon the original BEiT by introducing a more refined tokenization process for masked image modeling. BEiT-v2 leverages a teacher-student framework, where the teacher network generates discrete tokens from image patches, and the student network learns to predict these tokens from masked image patches. This approach enhances the model's ability to capture fine-grained visual patterns and contextual relationships. At its core, BEiT-v2 combines masked image modeling with a new tokenization strategy to achieve state-of-the-art performance on image classification and downstream tasks. In our work, we evaluate the ViT-B/16 architecture trained on ImageNet-1k using BEiT-v2.

iBOT. Zhou *et al*. [71] combine ideas from DINO and MAE by training a model to reconstruct masked dense features based on a teacher network. iBOT uses both an imagelevel and a dense distillation objective. We analyze the ViT-B/16 and ViT-L/16 architectures trained on ImageNet1k and ImageNet-22k. We evaluate the checkpoints released by the authors.

B. Task-Specific Metric Descriptions

Generic Object Segmentation We report the full results in Tab. 6 using the following metrics to evaluate generic object segmentation, which involves binary segmentation of foreground objects and background:

- **F1 Score:** The F1 score provides a harmonic mean of precision and recall, offering a balanced evaluation of segmentation performance, particularly in the presence of class imbalance. It is defined as:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

  where **Precision** measures the proportion of correctly predicted foreground pixels among all pixels predicted as foreground, and **Recall** measures the proportion of correctly predicted foreground pixels relative to all ground truth foreground pixels.
- **Accuracy:** Accuracy quantifies the proportion of correctly classified pixels, encompassing both foreground and background classes. It is defined as:

$$\text{Accuracy} = \frac{\text{Correct Predictions} \cdot (\text{Fore.} + \text{Back.})}{\text{Total Pixels}}$$

  While simple and intuitive, accuracy may be biased toward the majority class (e.g., background), particularly in cases of class imbalance.
- **Mean Intersection over Union (mIoU):** mIoU assesses segmentation performance by averaging the Intersection over Union (IoU) across all classes (foreground and background). For a given class c, IoU is defined as:

$$\text{IoU}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c}$$

  where TP_c, FP_c, and FN_c denote the true positives, false positives, and false negatives for class c. mIoU is computed as:

$$\text{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \text{IoU}_c$$

  where $C = 2$ for generic object segmentation. mIoU provides a robust evaluation of the model's capacity to capture spatial overlap and resolve fine-grained boundaries.

These metrics collectively provide a comprehensive evaluation of the model's performance in binary segmentation tasks, highlighting both pixel-level accuracy and the model's ability to distinguish between foreground and background regions.

Depth Prediction We present the complete results for depth prediction in Tab. 7. To evaluate performance, we adopt the setup described in [16], which includes computing the root mean square error (RMSE) and evaluating the prediction accuracy under different threshold criteria. The threshold-based accuracy, denoted as δ_i, measures the proportion of pixels for which the ratio between the predicted depth (d^{pr}) and the ground-truth depth (d^{gt}) lies below 1.25^i. Formally, this is defined as:

$$\delta_i(d^{pr}, d^{gt}) = \frac{1}{N} \sum_{j=1}^{N} \left[\max \left(\frac{d_j^{pr}}{d_j^{gt}}, \frac{d_j^{gt}}{d_j^{pr}} \right) < 1.25^i \right] \quad (1)$$

where N is the total number of pixels, d^{pr} represents the predicted depth, and d^{gt} is the ground-truth depth.

Surface Normal Estimation For each pixel in the image, the error is defined as the angular deviation (in degrees) between the predicted and ground-truth surface normals. To evaluate the model's performance, we compute two primary metrics: (1) the root mean square error (RMSE), which measures the overall angular error, and (2) the accuracy of predictions at predefined angular thresholds. Specifically, the accuracy metric is calculated as the proportion of pixels whose angular error falls within thresholds of $11.25°$, $22.5°$, and $30°$, following established evaluation protocols [2, 22, 49].

Geometric Correspondence We report full results on object geometric correspondence in Tab. 9 and scene geometric correspondence in Tab. 10. Correspondences are evaluated using either 2D projection error or 3D metric error. For

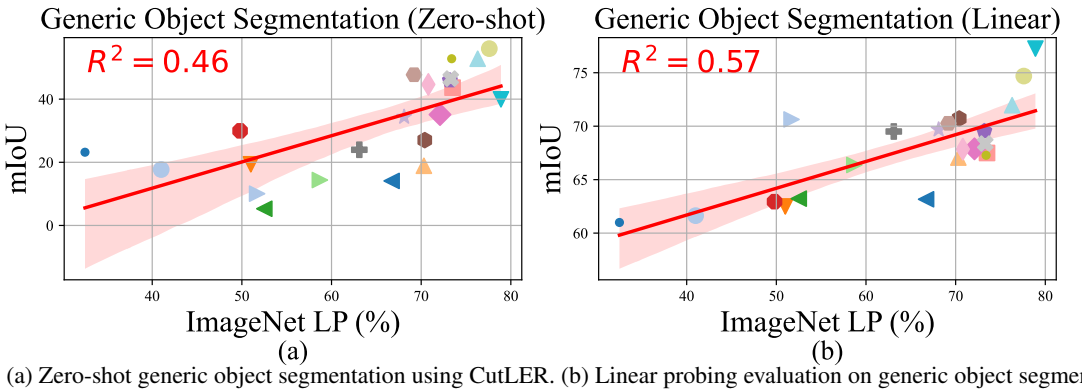

Figure 7. (a) Zero-shot generic object segmentation using CutLER. (b) Linear probing evaluation on generic object segmentation.

a correspondence between pixel locations p in image 1 and q in image 2, the 2D projection error is computed as follows. First, p is projected into 3D space, yielding a 3D point $\mathbf{P}$, using the depth value at p and the camera intrinsics of image 1. The 3D point $\mathbf{P}$ is transformed to the coordinate frame of image 2 using the relative camera pose and projected back onto the image plane of image 2, yielding the pixel location p'. The 2D projection error is then defined as:

$$\text{Error}_{2D} = \|p' - q\|_2$$

where $\|\cdot\|_2$ represents the Euclidean distance in the image plane.

For 3D metric error, both p and q are transformed into a shared 3D coordinate space, resulting in $\mathbf{P}$ and $\mathbf{Q}$, respectively. The 3D metric error is then computed as:

$$\text{Error}_{3D} = \|\mathbf{P} - \mathbf{Q}\|_2$$

The 2D projection error is used for scene-level correspondences, while the 3D metric error is preferred for objects to better account for occlusions and thin structures.

To evaluate correspondence quality, we compute *correspondence recall*, defined as the percentage of correspondences with error below a threshold τ:

$$\text{Recall} = \frac{|\{\text{Error} < \tau\}|}{N}$$

Where $|\{\text{Error} < \tau\}|$ indicates the number of correspondences with error below the threshold τ and N is the total number of correspondences. We report recall values for various τ values and analyze results across image pairs grouped by relative viewpoint changes.

Mid-level Image Similarity We present the full results for mid-level image similarity in Tab. 11. In this task, a reference image is provided, and the model selects one of two candidate images based on mid-level image similarity. The evaluation metrics used are Accuracy (Acc), Precision (Prec), Recall (Rec), and F1 Score (F1), defined as follows:
Accuracy (Acc): The proportion of correctly predicted matches out of the total comparisons:

$$\text{Acc} = \frac{\text{Correct Predictions}}{\text{Total Comparisons}}$$

Precision (Prec): The proportion of correctly identified matches (true positives, TP) among all images predicted as matches:

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{False Positives (FP)}}$$

Recall (Rec): The proportion of correctly identified matches (TP) among all actual matches in the dataset:

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{False Negatives (FN)}}$$

F1 Score (F1): The harmonic mean of Precision and Recall, providing a balanced measure of performance:

$$\text{F1} = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}}$$

These metrics provide a rigorous evaluation of the model's ability to identify mid-level image similarities accurately and consistently.

C. Potential Impact of DPT Decoder

Zero-shot vs. Fine-tuning: Fig. 7 provides a zero-shot evaluation on generic object segmentation using CutLER [60]. The observation aligns with our fine-tuning experiments — the R^2 value remains high. However, CutLER requires model-specific hyperparameters and achieves lower performance than fine-tuning DPT with frozen backbones.
Potential Impact of DPT Decoder: We ablate the impact of DPT decoder in (b) from Fig. 7. We conducted 22 additional experiments on generic object segmentation using linear probing, as shown in above figure (b). The overall trend and relative model rankings remain consistent with our findings using DPT.

D. Qualitative Comparisons

We present qualitative visualizations in Fig. 8 and Fig. 9 to assess model performance on mid-level vision tasks. These visualizations validate the models' ability to learn and perform each mid level vision task effectively.

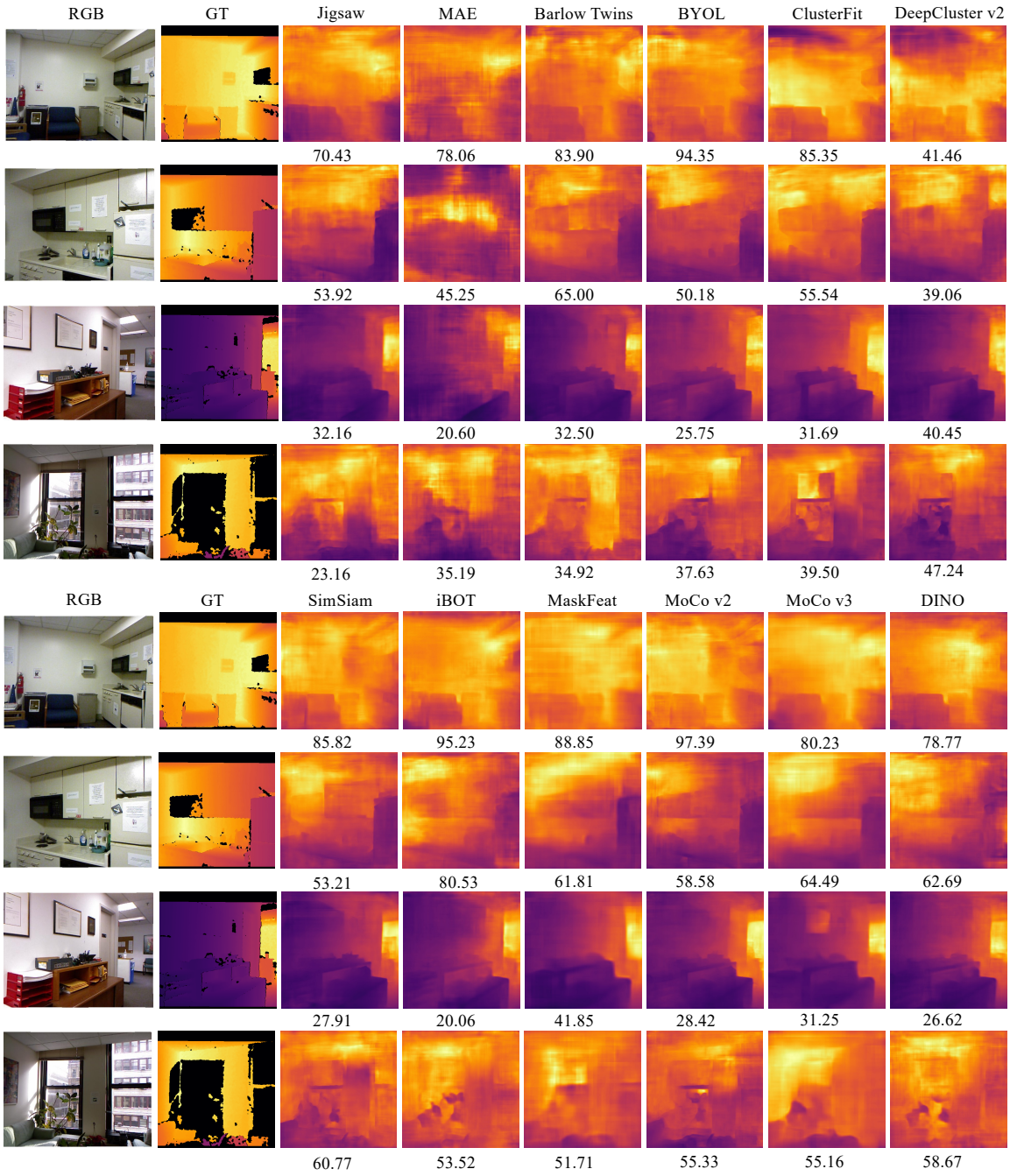

Figure 8. **Qualitative Depth Estimation Results for Selected SSL Models.** Depth estimation visualizations are shown for selected SSL models, with the δ_1 score displayed below each visualization (*higher is better*). These results highlight the models' effectiveness in capturing depth information. Note DINO and MoCo v3 are ViT based.

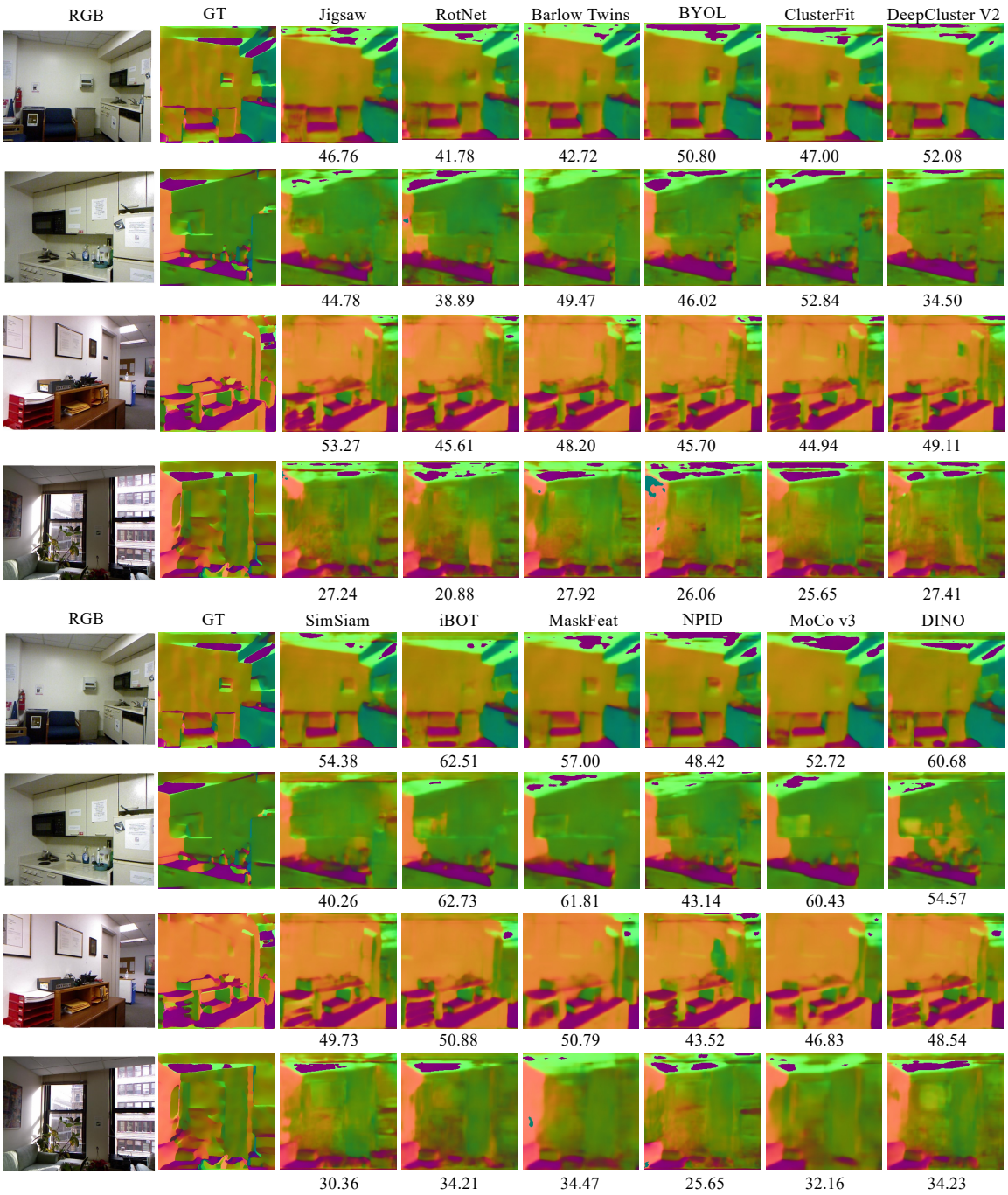

Figure 9. **Qualitative Surface Normal Estimation Results for Selected SSL Models.** Surface normal estimation visualizations are shown for selected SSL models, with the δ_1 score displayed below each visualization (*higher is better*). These results highlight the models' effectiveness in capturing surface normal information. Note DINO and MoCo v3 are ViT based.

Table 6. **2D Grouping Results (Generic Object Segmentation).** Evaluation results for generic object segmentation, where models segment foreground objects from the background, are presented for both VOC07 [19] and VOC12 [20] datasets.

Model	Backbone	Task	VOC07 [19]			VOC12 [20]		
			F1-measure	mIoU	Accuracy	F1-measure	mIoU	Accuracy
Self-Supervised Models (SSL)								
Jigsaw [45]	RN-50	IN-1k	71.13	63.03	83.24	81.51	71.48	89.41
RotNet [25]	RN-50	IN-1k	75.84	65.32	85.39	83.46	71.46	89.94
NPID [66]	RN-50	IN-1k	76.92	66.38	85.99	84.34	72.66	90.35
SeLa-v2 [4]	RN-50	IN-1k	83.20	73.53	89.73	86.03	76.56	91.71
NPID++ [42]	RN-50	IN-1k	80.75	69.59	87.84	85.46	75.24	91.29
PIRL [42]	RN-50	IN-1k	79.55	69.62	87.69	86.40	77.39	92.46
ClusterFit [67]	RN-50	IN-1k	77.91	67.94	86.79	85.58	72.98	90.25
DeepCluster-v2 [4]	RN-50	IN-1k	79.33	71.08	88.14	88.29	79.91	93.01
SwAV [4]	RN-50	IN-1k	79.72	71.95	88.59	87.38	78.72	92.91
SimCLR [6]	RN-50	IN-1k	81.05	73.63	89.44	87.94	79.62	93.25
MoCo v2 [8]	RN-50	IN-1k	82.78	74.40	89.91	88.65	79.75	93.21
SimSiam [7]	RN-50	IN-1k	82.99	74.05	89.88	88.25	77.51	92.05
BYOL [27]	RN-50	IN-1k	83.20	71.97	89.21	87.74	78.81	93.09
Barlow Twins [69]	RN-50	IN-1k	79.97	71.53	88.51	88.09	78.62	92.82
DenseCL [58]	RN-50	IN-1k	79.32	70.71	88.03	87.19	78.75	92.47
DINO [5]	RN-50	IN-1k	78.13	71.95	88.32	88.81	79.86	92.99
MoCo v3 [9]	RN-50	IN-1k	82.56	71.48	88.88	85.44	77.41	92.06
DINO [5]	ViT-B/16	IN-1k	83.12	74.00	89.79	88.70	79.94	93.17
iBOT [71]	ViT-B/16	IN-1k	82.85	75.74	90.50	90.51	84.72	94.90
MoCo v3 [9]	ViT-B/16	IN-1k	80.92	72.45	88.99	82.11	74.11	90.71
MAE [30]	ViT-B/16	IN-1k	77.25	65.78	85.88	80.22	69.63	89.14
MaskFeat [62]	ViT-B/16	IN-1k	78.84	70.28	87.76	84.27	75.14	91.00

Table 7. **Depth Estimation Results for SSL Models on NYU and NAVI.** Results for scene-level (NYU) and object-level (NAVI) depth estimation using self-supervised models. These results demonstrate the performance of SSL models across diverse depth estimation tasks.

Model	Architecture	Dataset	NYU				NAVI			
			δ_1	δ_2	δ_3	RMSE	δ_1	δ_2	δ_3	RMSE
Self-Supervised Models										
Jigsaw [45]	RN-50	IN-1k	71.17	93.02	98.24	0.6282	29.48	55.45	73.66	0.1775
RotNet [25]	RN-50	IN-1k	73.18	93.41	98.23	0.6047	29.87	55.03	73.00	0.1804
NPID [66]	RN-50	IN-1k	70.65	92.81	98.34	0.6191	37.88	65.46	80.82	0.1506
Sela-v2 [4]	RN-50	IN-1k	74.76	94.47	98.80	0.5684	34.72	61.97	78.64	0.1586
NPID++ [42]	RN-50	IN-1k	71.89	93.27	98.34	0.6110	38.07	65.32	80.69	0.1525
PIRL [42]	RN-50	IN-1k	74.58	94.13	98.59	0.5780	38.55	65.36	80.86	0.1495
ClusterFit [67]	RN-50	IN-1k	74.13	93.81	98.25	0.5850	39.45	66.47	81.45	0.1479
DeepCluster-v2 [4]	RN-50	IN-1k	73.63	93.62	98.39	0.5863	39.50	67.35	82.43	0.1448
SwAV [4]	RN-50	IN-1k	76.17	94.96	98.81	0.5542	39.45	67.13	82.04	0.1457
SimCLR [6]	RN-50	IN-1k	75.64	94.67	98.65	0.5698	42.86	70.04	83.68	0.1365
MoCo v2 [8]	RN-50	IN-1k	77.05	94.83	98.77	0.5467	45.42	72.55	85.42	0.1309
SimSiam [7]	RN-50	IN-1k	75.95	94.74	98.78	0.5628	43.03	70.01	83.94	0.1366
BYOL [27]	RN-50	IN-1k	75.43	94.48	98.68	0.5711	42.19	69.22	83.54	0.1387
Barlow Twins [69]	RN-50	IN-1k	75.06	94.22	98.61	0.5791	41.83	68.74	83.01	0.1408
DenseCL [58]	RN-50	IN-1k	76.30	94.69	98.65	0.5615	43.78	71.45	85.01	0.1332
DINO [5]	RN-50	IN-1k	77.68	95.89	99.09	0.5235	47.63	74.31	86.54	0.1241
MoCo v3 [9]	RN-50	IN-1k	75.56	94.63	98.86	0.5584	45.93	72.87	85.57	0.1309
DINO [5]	ViT-B/16	IN-1k	79.38	95.97	99.05	0.5278	47.75	74.65	87.02	0.1241
iBOT [71]	ViT-B/16	IN-1k	81.32	96.90	99.34	0.4919	50.02	76.29	87.89	0.1199
MoCo v3 [9]	ViT-B/16	IN-1k	80.14	96.14	99.16	0.5109	51.07	76.96	87.95	0.1175
MAE [30]	ViT-B/16	IN-1k	66.17	90.38	97.37	0.6898	26.78	51.82	71.69	0.1868
MaskFeat [62]	ViT-B/16	IN-1k	80.39	96.18	99.07	0.5125	49.50	75.47	87.14	0.1195

Table 8. **Surface Normal Estimation Results on NYUv2 and NAVI Datasets.** Performance of self-supervised models on scene-level (NYUv2) and object-level (NAVI) surface normal estimation, evaluated using angular thresholds (11.25°, 22.5°, 30°) and RMSE metrics.

			NYUv2				NAVI			
Model	**Backbone**	**Dataset**	**11.25°**	**22.5°**	**30°**	**RMSE**	**11.25°**	**22.5°**	**30°**	**RMSE**
Self-Supervised Models										
Jigsaw [45]	RN-50	IN-1k	44.27	67.65	76.23	28.8386	22.79	49.22	62.50	36.6169
RotNet [25]	RN-50	IN-1k	43.93	67.40	76.07	28.8557	23.70	50.20	63.46	28.8557
NPID [66]	RN-50	IN-1k	40.80	64.68	73.97	35.4511	24.92	51.82	64.87	35.4511
SeLa-v2 [4]	RN-50	IN-1k	45.14	68.98	77.53	28.0449	25.73	53.19	66.22	34.7204
NPID++ [42]	RN-50	IN-1k	41.57	65.98	75.14	29.2829	25.03	52.03	65.20	34.9940
PIRL [42]	RN-50	IN-1k	44.92	68.35	76.71	28.5771	27.01	54.06	66.85	34.1514
ClusterFit [67]	RN-50	IN-1k	43.93	67.40	76.12	28.9261	25.49	53.21	65.98	34.8134
Deepcluster-v2 [4]	RN-50	IN-1k	44.48	68.29	76.98	28.2509	26.51	54.01	67.07	34.1514
SwAV [4]	RN-50	IN-1k	44.08	67.98	76.81	28.2881	25.69	53.17	66.21	34.4863
SimCLR [6]	RN-50	IN-1k	45.87	69.17	77.48	27.9438	26.70	54.21	67.07	34.1743
MoCo v2 [8]	RN-50	IN-1k	46.37	69.79	78.03	27.5874	29.02	56.86	69.42	32.7033
SimSiam [7]	RN-50	IN-1k	44.12	67.95	76.72	28.4032	28.06	55.71	68.18	33.5474
BYOL [27]	RN-50	IN-1k	43.64	67.73	76.46	28.5432	26.51	54.29	67.17	34.1015
Barlow Twins [69]	RN-50	IN-1k	44.04	67.75	76.57	28.4161	27.21	54.70	67.46	33.9390
DenseCL [58]	RN-50	IN-1k	45.30	68.74	77.16	28.2974	27.21	54.70	67.46	33.9390
DINO [5]	RN-50	IN-1k	47.64	70.96	79.12	26.8891	31.43	59.50	71.77	31.3895
MoCo v3 [9]	RN-50	IN-1k	43.03	67.15	76.20	28.6994	27.22	55.03	67.85	33.7240
DINO [5]	ViT-B/16	IN-1k	48.42	69.71	77.57	28.0873	31.66	58.58	70.68	31.9912
iBOT [71]	ViT-B/16	IN-1k	52.02	72.43	79.53	26.9539	32.75	60.06	71.69	31.4563
MoCo v3 [9]	ViT-B/16	IN-1k	49.64	70.01	77.36	28.2596	31.72	57.84	69.20	33.0295
MAE [31]	ViT-B/16	IN-1k	43.89	66.13	74.56	30.1382	22.07	49.06	62.63	36.4724
MaskFeat [62]	ViT-B/16	IN-1k	53.63	72.23	79.03	27.1797	32.40	58.92	70.43	32.3430

Table 9. **Geometric Correspondence Results on the NAVI Dataset.** Evaluation of self-supervised models on geometric correspondence tasks, including 3D Recall, 2D Projection Recall, and Binned Recall, across varying viewpoint angle ranges.

			3D Recall			2D Recall			Bin Recall			
Model	**Architecture**	**Dataset**	0.01m	0.02m	0.05m	5px	25px	50px	0-30°	30-60°	60-90°	90-120°
Self-Supervised Models (SSL)												
Jigsaw [45]	RN-50	IN-1k	9.13	19.83	54.94	0.68	7.45	16.20	49.15	26.54	13.06	7.76
RotNet [25]	RN-50	IN-1k	11.97	23.21	55.13	0.92	9.83	19.38	58.44	29.82	14.85	10.26
NPID [66]	RN-50	IN-1k	18.70	32.11	63.38	1.57	15.80	27.47	69.09	41.51	22.96	16.62
SeLa v2 [4]	RN-50	IN-1k	12.17	23.50	53.26	0.93	10.14	19.49	49.33	28.07	18.86	12.86
NPID++ [42]	RN-50	IN-1k	13.20	25.86	58.25	0.87	10.52	21.20	53.10	32.17	19.75	14.41
PIRL [42]	RN-50	IN-1k	16.21	29.49	61.54	1.15	13.21	24.73	60.73	36.56	22.61	16.40
ClusterFit [67]	RN-50	IN-1k	10.85	21.49	56.86	1.86	9.08	16.94	43.28	26.57	17.32	11.61
DeepCluster v2 [4]	RN-50	IN-1k	20.65	34.42	64.24	1.78	18.14	30.46	69.52	42.24	27.47	19.09
SwAV [4]	RN-50	IN-1k	20.20	33.99	63.20	1.71	17.60	29.83	67.11	42.34	27.23	18.81
SimCLR [6]	RN-50	IN-1k	16.57	30.68	61.75	1.09	13.49	25.80	60.53	37.77	23.67	18.27
MoCo v2 [8]	RN-50	IN-1k	21.85	37.76	68.76	1.63	18.17	32.94	75.85	48.73	28.47	20.50
SimSiam [7]	RN-50	IN-1k	23.47	38.16	68.41	2.07	20.16	33.57	76.05	48.63	29.90	20.46
BYOL [27]	RN-50	IN-1k	10.81	21.11	56.81	2.26	9.02	16.64	46.24	26.45	15.82	10.65
Barlow Twins [69]	RN-50	IN-1k	12.71	23.27	58.22	2.97	10.92	18.83	52.25	29.38	17.00	11.41
DenseCL [58]	RN-50	IN-1k	17.59	34.57	67.63	1.17	14.28	29.17	71.25	44.65	26.29	17.76
DINO [5]	RN-50	NAVI	30.57	47.36	75.43	2.61	26.79	42.41	84.37	61.43	39.01	26.82
MoCo v3 [9]	RN-50	NAVI	21.70	36.29	65.49	1.70	18.43	31.77	73.41	45.90	27.84	19.88
DINO [5]	ViT-B/16	IN-1k	25.91	43.00	74.66	3.16	22.54	36.86	84.78	56.28	33.20	22.54
iBOT [71]	ViT-B/16	IN-1k	26.84	44.72	76.10	3.12	23.78	39.11	86.94	58.98	34.22	23.85
MoCo v3 [9]	ViT-B/16	IN-1k	26.99	44.46	75.22	2.17	23.45	39.54	85.95	58.96	34.45	23.20
MAE [30]	ViT-B/16	IN-1k	19.21	32.59	66.82	2.74	17.16	27.72	78.17	46.12	21.16	11.85
MaskFeat [62]	ViT-B/16	IN-1k	22.11	35.16	65.92	2.08	19.67	31.37	86.25	51.50	22.17	11.00

Table 10. **Geometric Correspondence Results on ScanNet.** Evaluation of self-supervised models on 2D Projection Recall at varying pixel error thresholds and Binned Recall across viewpoint angle ranges.

		2D Recall			Bin Recall			
Model	**Architecture**	5px	10px	20px	0-15°	15-30°	30-60°	60-180°
Self-Supervised Models								
Jigsaw [45]	RN-50	9.57	18.18	27.98	26.11	19.80	11.16	4.00
RotNet [25]	RN-50	15.74	25.46	34.15	37.56	28.52	13.73	4.29
NPID [66]	RN-50	27.64	40.10	50.07	52.84	44.85	28.24	11.34
SeLa-v2 [4]	RN-50	12.21	22.70	33.36	31.73	24.61	14.61	6.54
NPID++ [42]	RN-50	10.62	19.59	30.23	27.16	20.92	13.09	6.37
PIRL [42]	RN-50	17.89	30.43	41.35	45.37	35.12	19.67	7.55
ClusterFit [67]	RN-50	26.31	40.92	51.96	54.96	46.61	26.67	10.45
DeepCluster v2 [4]	RN-50	17.30	27.90	37.57	38.25	30.90	18.09	7.87
SwAV [4]	RN-50	25.41	38.74	49.86	52.34	44.20	27.48	10.23
SimCLR [6]	RN-50	21.78	35.34	46.18	48.85	40.32	22.15	9.08
MoCo v2 [8]	RN-50	24.92	37.65	48.33	50.92	41.97	24.56	8.97
SimSiam [7]	RN-50	18.11	29.83	40.92	42.58	33.72	19.04	7.24
BYOL [27]	RN-50	15.39	25.41	34.89	35.88	26.91	16.96	6.90
Barlow Twins [69]	RN-50	18.83	30.60	40.61	42.36	33.96	19.24	8.55
DenseCL [58]	RN-50	17.23	31.17	44.98	42.41	34.80	20.36	8.56
DINO [5]	RN-50	26.63	40.64	51.49	54.07	45.63	27.80	11.19
MoCo v3 [9]	RN-50	15.23	26.06	35.87	37.24	28.23	15.94	7.05
DINO [5]	ViT-B/16	24.38	34.22	45.47	46.56	36.72	23.74	11.12
iBOT [71]	ViT-B/16	20.04	29.45	41.07	41.13	30.95	20.00	9.47
MoCo v3 [9]	ViT-B/16	25.03	39.31	51.00	53.18	42.87	27.05	11.95
MAE [31]	ViT-B/16	6.64	10.31	18.42	15.64	9.81	6.63	3.81
MaskFeat [62]	ViT-B/16	27.94	40.87	50.49	56.51	47.65	24.41	6.90

Table 11. **Image Retrieval Results on the NIGHTS Dataset.** Retrieval performance is evaluated using Accuracy, F1-score, Precision, and Recall. Results are reported for self-supervised models using ResNet-50 and ViT-B/16 backbones, highlighting their capability to retrieve similar images based on mid-level features.

Model	**Backbone**	**Dataset**	Accuracy	F1-Score	Precision	Recall
Self-Supervised Models (SSL)						
Jigsaw [45]	RN-50	NIGHTS	71.22	70.69	70.73	70.65
RotNet [25]	RN-50	NIGHTS	75.33	75.14	74.40	75.89
NPID [66]	RN-50	NIGHTS	81.41	81.16	80.84	81.47
SeLa-v2 [4]	RN-50	NIGHTS	81.41	81.16	80.84	81.47
NPID++ [42]	RN-50	NIGHTS	83.06	82.63	83.24	82.03
PIRL [42]	RN-50	NIGHTS	83.77	83.56	83.19	83.93
ClusterFit [67]	RN-50	NIGHTS	81.58	81.42	80.70	82.14
DeepCluster-v2 [4]	RN-50	NIGHTS	85.25	84.93	85.26	84.60
SwAV [4]	RN-50	NIGHTS	84.65	84.36	84.45	84.26
SimCLR [6]	RN-50	NIGHTS	83.55	83.26	83.26	83.26
MoCo v2 [8]	RN-50	NIGHTS	84.43	84.22	83.85	84.60
SimSiam [7]	RN-50	NIGHTS	85.86	85.78	84.75	86.83
BYOL [27]	RN-50	NIGHTS	85.86	85.75	84.90	86.61
Barlow Twins [69]	RN-50	NIGHTS	83.11	82.70	83.26	82.14
DenseCL [58]	RN-50	NIGHTS	82.73	82.53	82.03	83.04
DINO [5]	RN-50	NIGHTS	83.83	83.31	84.50	82.14
MoCo v3 [9]	RN-50	NIGHTS	84.70	84.37	84.70	84.04
DINO [5]	ViT-B/16	NIGHTS	89.20	88.98	89.23	88.73
iBOT [71]	ViT-B/16	NIGHTS	89.36	89.27	88.49	90.07
MoCo v3 [9]	ViT-B/16	NIGHTS	87.17	86.90	87.19	86.61
MAE [30]	ViT-B/16	NIGHTS	83.39	82.91	83.81	82.03
MaskFeat [62]	ViT-B/16	NIGHTS	76.10	75.70	75.61	75.78