

FisherMatch: Semi-Supervised Rotation Regression via Entropy-based Filtering

Yingda Yin Yingcheng Cai He Wang[†] Baoquan Chen[†]
Peking University

Abstract

Estimating the 3DoF rotation from a single RGB image is an important yet challenging problem. Recent works achieve good performance relying on a large amount of expensive-to-obtain labeled data. To reduce the amount of supervision, we for the first time propose a general framework, FisherMatch, for semi-supervised rotation regression, without assuming any domain-specific knowledge or paired data. Inspired by the popular semi-supervised approach, FixMatch, we propose to leverage pseudo label filtering to facilitate the information flow from labeled data to unlabeled data in a teacher-student mutual learning framework. However, incorporating the pseudo label filtering mechanism into semi-supervised rotation regression is highly non-trivial, mainly due to the lack of a reliable confidence measure for rotation prediction. In this work, we propose to leverage matrix Fisher distribution to build a probabilistic model of rotation and devise a matrix Fisher-based regressor for jointly predicting rotation along with its prediction uncertainty. We then propose to use the entropy of the predicted distribution as a confidence measure, which enables us to perform pseudo label filtering for rotation regression. For supervising such distribution-like pseudo labels, we further investigate the problem of how to enforce loss between two matrix Fisher distributions. Our extensive experiments show that our method can work well even under very low labeled data ratios on different benchmarks, achieving significant and consistent performance improvement over supervised learning and other semi-supervised learning baselines. Our project page is at <https://yd-yin.github.io/FisherMatch>.

1. Introduction

Incorporating deep neural networks to perform rotation regression is exerting an ever-important influence in computer vision, graphics and robotics. This is now one of the key technology in enabling a multitude of applications such as camera relocalization and visual odometry [8, 14], ob-

ject pose estimation and tracking [48, 53], and 6DoF robot grasping [7, 20]. One of the major obstacles to improving rotation regression is expensive rotation annotations. Though many large-scale image datasets have been curated with sufficient semantic annotations, obtaining a large-scale real dataset with rotation annotations can be extremely laborious, expensive and error-prone [52]. With the amount of labeled data being the bottleneck, there is a demand for methods that can leverage unlabeled data.

Regarding training models with fewer labels, semi-supervised learning (SSL) has been a powerful approach, mitigating the requirement for labeled data by providing a means of leveraging unlabeled data and thus attracting more and more attention.

Recent years have witnessed many processes in semi-supervised classification [5, 15, 26, 43, 44], semi-supervised object detection [32, 47], and semi-supervised human and hand pose estimation [22, 39]. However, only few works address semi-supervised rotation regression, with most of them leveraging domain-specific knowledge, e.g., temporal smoothness of object pose [31] and strong assumptions, e.g., paired images from different viewpoints [34].

The underlying reason for little work in this field is that rotation regression is very unique and challenging. First, it is undesirable to turn the rotation regression into a classification problem. Given that 3D rotation space is continuous, discretizing the space into a small number of bins will lead to limited accuracy, which is intolerable for many applications involving rotation estimation. Also, rotation regression is even not a standard regression problem. Given that rotation space $SO(3)$ is a non-Euclidean manifold [60], a general regression algorithm needs to be tailored, taking the nonlinear structure of the rotation space into account. This further makes semi-supervised rotation regression a more challenging and less studied topic.

In this work, for the first time, we propose a general framework, namely *FisherMatch*, for semi-supervised rotation regression. The problem we tackle is very general: *using a neural network to regress rotation from a single RGB image*. Inspired by a popular semi-supervised learning approach, FixMatch [43], initially developed for classification tasks, we attempt to process rotation regression problems in

[†]He Wang and Baoquan Chen are the corresponding authors ({hewang, baoquan}@pku.edu.cn).

a similar flavor.

The key idea to the success of FixMatch is *to filter out the pseudo labels with low classification confidence and only supervise the model outputs with highly confident labels*. This mechanism ensures the quality of pseudo labels and thus significantly improves the performance of semi-supervised learning. The underlying assumption is that the more confident a pseudo label is, the more closed this label is to the ground truth. Or, in other words, this system needs to predict a confidence that can well indicate the correctness of its prediction. Fortunately, a classification output naturally carries the information: the probability of its prediction can be used as its prediction confidence. We argue that the availability of such a reliable confidence measure is crucial to the success of FixMatch on semi-supervised classification tasks. Similarly, when adopting FixMatch to 3D object detection, 3DIoUMatch [47] constructs a separate branch to predict the 3D IoU between the predicted bounding box and the ground truth bounding box as a localization confidence to filter out poor predictions. Although 3D IoU estimation is a regression task, 3DIoUMatch can move around the predicted bounding boxes as an augmentation trick, thus creating an infinite amount of training data for this confidence estimation module. This augmentation is crucial for such a confidence estimation module since the confidence estimation modules can only be trained using labeled data and must work on unlabeled data.

However, we argue that adopting FixMatch for rotation estimation is highly non-trivial. The biggest obstacle is how to estimate the prediction confidence for rotation regression. For rotation regression, we don't have the largest probability from the bins as our confidence; also, for rotation estimation from a single RGB image, we can't perform such augmentation to change our rotation prediction; yet, we still need this uncertainty estimation module to work on unlabeled data with only training on a small set of labeled data.

As pointed out by [42], probabilistic modeling of rotation is the correct way to model the uncertainty of rotation regression. Parametric statistical methods for orientation statistics have long been established [12, 21, 24, 40]. In order to better resort to $SO(3)$ manifold which has a different topology than unconstrained values in \mathbb{R}^N , Deng *et al.* [10] and Mohlin *et al.* [35] incorporate Bingham distribution and matrix Fisher distribution respectively to automatically learn uncertainties along with predictions, without further supervision. Thus, such networks can provide valuable information about the quality of the prediction. We prefer matrix Fisher distribution to Bingham distribution, since its rotation representation is continuous and its loss is convex with bounded gradient magnitudes, resulting in a stable training for neural networks [29, 35].

We thus devise a matrix Fisher-based rotation regressor that takes input a single RGB image and outputs the param-

eter of a matrix-Fisher distribution. Given the predicted distribution, we propose to use the entropy of this distribution as a confidence measure for pseudo label filtering. Basically, only pseudo labels with high confidence, *i.e.* lower entropy than a threshold τ_{entropy} , will pass the filtering and be used for supervising the model under training. Our experiment consistently proves that entropy is an efficient indicator of the prediction performance, not only in the case of 100 percent labeled data, but also in low data ratio cases down to 5 percent. Since FisherMatch outputs a distribution rather than a single rotation, our pseudo labels become a distribution, which requires research into the unsupervised loss enforced between two distributions. In this work, we investigate cross entropy loss and negative log likelihood loss, draw a connection between them, and find their proper usage in our experiments.

On common benchmark datasets of object rotation estimation from RGB images (ModelNet10-SO(3) and Pascal3D+) under various labeled data ratios, our experiment demonstrates a significant and consistent performance improvement over supervised learning and other semi-supervised learning baselines.

2. Related Work

Rotation regression The choice of rotation representation is one of the core issues concerning rotation regression. The commonly used representations include Euler angles, axis-angles, unit quaternions, *etc.* However, Euler angles suffer from gimbal lock, and quaternions have a double embedding giving rise to the existence of two disconnected local minima. Moreover, [60] argues that representations less than 4 dimensions are bound to have discontinuities and are difficult for neural networks to learn. To this end, the continuous 6D representation with Gram-Schmidt orthogonalization [60] and 9D representation with SVD orthogonalization [29] have been proposed respectively, leading to superior performance in rotation regression.

Several works propose to use probability distributions over rotations to further model prediction uncertainties along with rotation regression. In Prokudin *et al.* [42], parameters of a mixture of Von Mises distribution using a biternion network are estimated. Deng *et al.* [10] uses Bingham distribution over unit quaternions to jointly predict the rotation as well as the uncertainty. Estimation with matrix Fisher distribution [35] learns to build the probability distribution over rotation matrices with unconstrained parameters. To further express arbitrary rotation distributions and better tackle rotation regression for symmetry objects, Implicit-PDF [36] chooses to represent the distributions implicitly by neural networks, instead of distribution parameters, where the $SO(3)$ space is uniformly discretized with the help of Hopf fibration [56].

Semi-supervised classification Semi-supervised learning is a long-studied field with a diversity of approaches, many in the field of classification. Consistency regularization and pseudo labeling are two measures with in-depth exploration. Consistency regularization was first proposed in [3] which enforces the model to predict consistently across multiple perturbations [23, 26, 44, 54]. Pseudo labels [27] are artificial labels generated by the model itself and are used to further train the model, often applied along with a confidence-based thresholding to ensure the pseudo label quality. Mixmatch [5], ReMixmatch [4] and FixMatch [43] are holistic methods utilizing various augmentation and label sharpening strategies.

More recently, SIMPLE [18] proposes the paired loss minimizing the statistical distance between confident and similar pseudo labels. SemCo [38] considers label semantics to prevent the degradation of pseudo label quality for visually similar classes in a co-training manner. Dash [55] and FlexMatch [58] propose dynamic and adaptive pseudo label filtering, better suited for the training process.

Semi-supervised regression Semi-supervised regression is a less-touched field compared with classification, where most of the works deal with regressing Euclidean variables, e.g., Parkinson’s disease rating scales from multiple telemonitoring data in UCI repository [2]. Early work of CoReg [61] utilizes multiple k-nearest neighbor regressors with different distance metrics and leverages the predictions of one regressor to label the other regressors in a co-training manner. SSDKL [19] leverages the unlabeled data by minimizing predictive variance in the posterior regularization framework through the composition of neural networks and the probabilistic modeling of Gaussian processes.

Self-/semi-supervised rotation estimation Several works tackle rotation estimation in a self-supervised manner. Mustikovela *et al.* [37] leverages the analysis-by-synthesis technique that requires a lot of extra images for training a generative model. ViewNet [34] assumes the availability of paired data (same object, different poses). The most relevant semi-supervised learning work is NVSM [46], which shares the same assumptions on data and labels with us. In contrary to *regression*, NVSM builds a category-level 3D cuboid mesh with feature vectors and estimates the object rotation in a render-and-compare technique through the distance-based rotation retrieval. Less literature has been seen in the field of semi-supervised rotation regression. Mariotti *et al.* [33] requires paired images of an object and enforces cross-reconstruction in an analysis-and-synthesis manner via rotating the encoded neural latent variables.

Our work draws insight from both the orientation statistics and the semi-supervised learning techniques introduced above, dedicated to correlating the techniques in two well-

explored fields to tackle the problem in the general setting of semi-supervised rotation regression.

3. Method

In this work, we tackle the problem of learning to predict 3D object rotation from single RGB images under a semi-supervised setting, where we have only a (small) set of labeled data $\{\mathbf{x}_i^l, \mathbf{y}_i^l\}_{i=1}^{N_l}$ and a larger set of unlabeled data $\{\mathbf{x}_i^u\}_{i=1}^{N_u}$. Here, \mathbf{x}^l and \mathbf{x}^u represent the labeled and unlabeled RGB image respectively, and \mathbf{y}^l represents the ground-truth rotation in $SO(3)$ for a labeled data; N_l and N_u are the number of labeled and unlabeled images, respectively.

Following a popular semi-supervised learning approach, FixMatch [43], we adopt the teacher-student mutual learning framework, which we summarize in Section 3.1. In Section 3.2, we make use of two probabilistic models of rotation for depicting the uncertainty in rotation prediction, namely Bingham distribution and matrix Fisher distribution [10, 35], and propose to use the entropy of the predicted matrix Fisher distribution as the prediction confidence for pseudo label filtering; In Section 3.4, for the purpose of enforcing loss between the teacher and the student, we construct two loss functions between pseudo labels and predicted distributions; Finally, in Section 3.5, we introduce our training protocol in detail.

3.1. Revisit FixMatch

The teacher-student mutual learning framework is a popular approach for semi-supervised learning. Mean Teacher [44] proposes the first version, containing two jointly learned models - a *teacher* and a *student*. The parameters of the teacher model are the exponential moving average (EMA) of the student model parameters that are updated by the stochastic gradient descent. The student model is trained by the ground-truth labels for the labeled data, and for the unlabeled data, the predictions of the teacher model serve as the *pseudo labels* and are used to supervise the student network, through which, a history consistency is enforced between the two models.

FixMatch [43] further develops this approach by proposing two strategies: asymmetric data augmentation and confidence-based pseudo label filtering. Asymmetric data augmentation means that the teacher model is fed by weakly augmented unlabeled samples while the student model takes strongly augmented unlabeled samples which contributes to the performance gap between the teacher and the student, facilitating correct information flow to the student.

Arguably, the most important contribution of FixMatch is to demonstrate the effectiveness of confidence-based pseudo label filtering. For a non-trivial semi-supervised learning task, previous works recognize that the pseudo la-

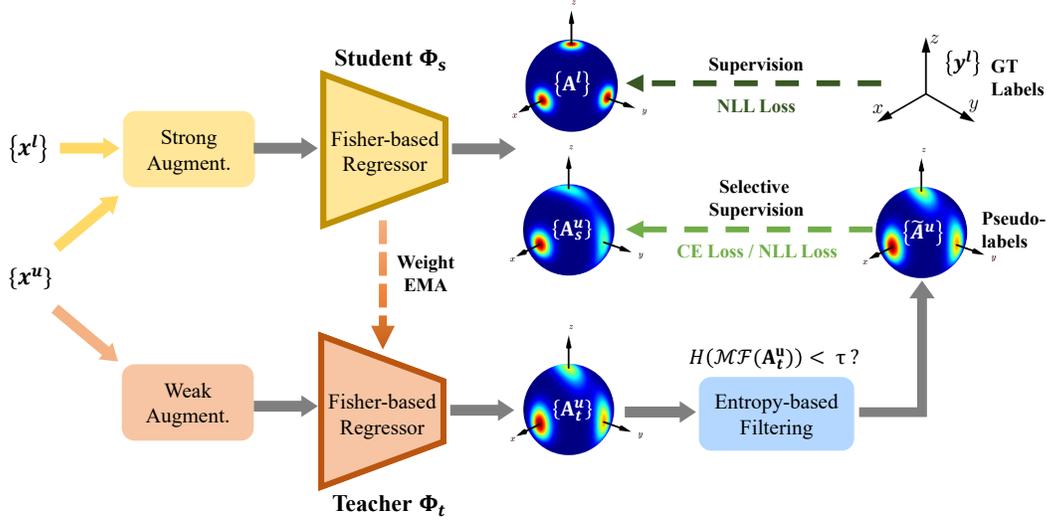


Figure 1. **Pipeline overview.** Our matrix Fisher based-rotation regressor Φ takes an RGB image \mathbf{x} as input and outputs the parameter \mathbf{A} of the predicted matrix Fisher distribution. We leverage a teacher-student mutual learning framework composed of a learnable student model and an exponential-moving-average (EMA) teacher model. On labeled data, the student network is trained by the ground-truth labels with the supervised loss; while on unlabeled data, the student model takes the pseudo labels from the EMA teacher. We leverage an entropy-based filtering technique to filter out noisy teacher predictions. The distribution visualization is borrowed from [35] where x, y and z shown in black axes correspond to the standard basis of \mathbb{R}^3 , and the pdf is shown on the sphere with a *jet* color coding. See Appendix Section D for details of the visualization method.

bels generated by the teacher output suffer from significant noises [43, 47]. To this end, FixMatch proposes to filter out low-quality predictions and only supervise the student model with predictions with high confidence. This strategy avoids wrong supervision to the student model and has been proved to be very effective for challenging tasks, *e.g.*, object detection [32, 47]. Given the difficulty of rotation regression, we further propose to leverage FixMatch as the basis of our framework for the rotation regression task.

3.2. Probabilistic Modeling of Rotation

To model the uncertainty of rotation estimation, we leverage *matrix Fisher* distribution to build a probabilistic model of rotation prediction, following Mohlin *et al.* [35].

Matrix Fisher distribution [24, 41] $\mathcal{MF}(\mathbf{R}; \mathbf{A})$ is a probability distribution over $\text{SO}(3)$ for rotation matrices, whose probability density function is in the form of

$$p(\mathbf{R}) = \mathcal{MF}(\mathbf{R}; \mathbf{A}) = \frac{1}{F(\mathbf{A})} \exp(\text{tr}(\mathbf{A}^T \mathbf{R})) \quad (1)$$

where parameter $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ is an arbitrary 3×3 matrix and $F(\mathbf{A})$ is the normalizing constant. The mode and dispersion of the distribution can be computed from computing singular value decomposition of the parameter \mathbf{A} . Assume $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and the singular values are sorted in descending order, the mode of the distribution is computed as

$$\hat{\mathbf{R}} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{UV}) \end{bmatrix} \mathbf{V}^T \quad (2)$$

and the singular values $\mathbf{S} = \text{diag}(s_1, s_2, s_3)$ indicates the strength of concentration. The larger a singular value s_i is, the more concentrated the distribution is along the corresponding axis (the i -th column of mode $\hat{\mathbf{R}}$).

Another important probabilistic model for rotation is *Bingham* distribution on \mathcal{S}^3 for unit quaternions. The probability density function is defined as

$$\mathcal{B}(\mathbf{q}; \mathbf{M}, \mathbf{Z}) = \frac{1}{F(\mathbf{Z})} \exp(\mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q}) \quad (3)$$

where $\mathbf{M} \in \text{O}(4)$ is a 4×4 orthogonal matrix and $\mathbf{Z} = \text{diag}(0, z_1, z_2, z_3)$ is a 4×4 diagonal matrix with $0 \geq z_1 \geq z_2 \geq z_3$. The first column of parameter \mathbf{M} indicates the mode and the remaining columns describe the orientation of dispersion while the corresponding z_i , ($i \in 1, 2, 3$) describe the strength of the dispersion. $F(\mathbf{Z})$ is the normalizing constant.

It is well recognized that rotation matrix \mathbf{R} and quaternion \mathbf{q} are two different representations of rotation. Similarly, as discussed in [41], matrix Fisher distribution and Bingham distribution are equivalent to each other differing only in parameterizations and rotation representations. However, given that quaternion is not a continuous representation of rotation [60], using matrix representation to learn a deep rotation estimation model has an intrinsic advantage and usually yields better performance. [35] further shows that matrix Fisher distribution has a bounded gradient, which is favored by deep neural networks. Therefore, 9D rotation matrix is chosen as our representation, and ma-

trix Fisher distribution is used for building our probabilistic rotation model.

3.3. Entropy-based Pseudo Label Filtering

Inspired by FixMatch, we only want the accurate predictions from the teacher model to “teach” the student model. Otherwise, noisy pseudo labels may slow down the training procedure, or even do harm to the whole process.

For depicting the confidence of a predicted distribution, we propose to use *entropy*, which is widely used in statistics acting as the degree of disorder or randomness in the system, as a measure of *uncertainty*. A lower entropy generally indicates a more *peaked* distribution which exhibits less uncertainty and higher confidence.

In this work, we propose an entropy-based filtering mechanism leveraging the probabilistic modeling of the rotation estimation over $SO(3)$. We devise a rotation regressor Φ that takes a single RGB image \mathbf{x} and outputs the parameter $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ of a matrix Fisher distribution

$$\mathbf{A} = \Phi(\mathbf{x}), \quad (4)$$

which not only contains a predicted rotation as the *mode* of this distribution, but also encode the information of the distribution *concentration*. We then compute the entropy of this predicted distribution (see Equation 9).

For *pseudo label filtering*, we set a fixed entropy threshold τ , and only reserve the prediction as a pseudo label if its entropy is lower than the threshold. Specifically, for unlabeled data \mathbf{x}^u , assume $p_t = \mathcal{MF}(\mathbf{A}_t^u)$ is the teacher output with $\mathbf{A}_t^u = \Phi_t(\mathbf{x}^u)$ and $p_s = \mathcal{MF}(\mathbf{A}_s^u)$ is the student output with $\mathbf{A}_s^u = \Phi_s(\mathbf{x}^u)$, the loss on unlabeled data is therefore:

$$L_u(\mathbf{x}^u) = \mathbb{1}(H(p_t) \leq \tau) L(p_t, p_s) \quad (5)$$

We discuss the loss function enforced between two distribution $L(p_t, p_s)$ in Section 3.4.

3.4. Loss Function between Distributions

For the labeled set $\{\mathbf{x}_i^l, \mathbf{y}_i^l\}_{i=1}^{N_l}$, we adopt the most common loss function, negative log likelihood (NLL) loss, to learn the probabilistic model of rotation, as in [10, 35]. This loss minimizes the negative log likelihood of the ground-truth rotation in the predicted distributions, as shown below:

$$L_l(\mathbf{x}^l, \mathbf{y}^l) = -\log(\mathcal{MF}(\mathbf{y}^l; \mathbf{A}^l)) \quad (6)$$

where $\mathbf{A}(\mathbf{x}^l)$ denotes the network output fed with input \mathbf{x}^l .

For unlabeled data, both our network predictions and pseudo labels are distributions, and thus we need to enforce loss between two distributions, which is rarely the case for a regression problem. We investigate two types of losses, *i.e.*, negative log likelihood (NLL) loss and cross entropy (CE) loss.

Cross Entropy Loss L^{CE} In classification problems, a widely-used loss function between two discrete distributions is cross entropy loss L^{CE} , whose gradient is equivalent to the gradient of KL divergence between two distribution [13]. We thus extend cross entropy loss L^{CE} so as to enforce the consistency between pseudo labels and the student outputs:

$$L^{\text{CE}}(p_t, p_s) = H(p_t, p_s) \quad (7)$$

To compute L^{CE} between two continuous distribution on $SO(3)$, we derive the analytical formula for the cross entropy between two matrix Fisher distributions $f \sim \mathcal{MF}(\mathbf{A}_f)$ and $g \sim \mathcal{MF}(\mathbf{A}_g)$, as shown below:

Assume $\mathbf{A}_f = \mathbf{U}_f \mathbf{S}_f \mathbf{V}_f^T$, $\mathbf{A}_g = \mathbf{U}_g \mathbf{S}_g \mathbf{V}_g^T$, γ is the standard transform from unit quaternion to rotation matrix, \mathbf{e}_i is the i -th column of \mathbf{I}_4 , and $\mathbf{E}_i = \gamma(\mathbf{e}_i)$, then we can derive

$$H(f, g) = \log F_g - \sum_{i=1}^4 z_{gi} \left(b_i^2 + \sum_{j=1}^4 (a_{ij}^2 - b_i^2) \frac{1}{F_f} \frac{\partial F_f}{\partial z_{fj}} \right) \quad (8)$$

$$\begin{aligned} \text{where } z_{gi} &= \text{tr}(\mathbf{E}_i^T \mathbf{S}_g) & z_{fj} &= \text{tr}(\mathbf{E}_j^T \mathbf{S}_f) \\ a_{ij} &= \gamma^{-1}(\mathbf{U}_f \mathbf{E}_i \mathbf{V}_f^T) \cdot \gamma^{-1}(\mathbf{U}_g \mathbf{E}_j \mathbf{V}_g^T) \\ b_i &= \gamma^{-1}(\mathbf{U}_f \mathbf{E}_i \mathbf{V}_f^T) \cdot \gamma^{-1}(\mathbf{U}_g \mathbf{E}_i \mathbf{V}_g^T) \end{aligned}$$

and F_f and F_g are constant wrt. parameter \mathbf{Z} . See Appendix Section B for the derivation. Note that when $f = g$, we can also get the entropy $H(f)$ for matrix Fisher distribution, as shown below:

$$H(f) = \log F_f - \sum_{i=1}^4 \left(z_{fi} \frac{1}{F_f} \frac{\partial F_f}{\partial z_{fi}} \right) \quad (9)$$

NLL Loss L^{NLL} Another option of the loss is to consider the negative log likelihood of the mode predicted by the teacher in the distribution predicted by the student, which is basically the NLL loss treating the teacher prediction as ground truth, as in the case of labeled data.

$$L^{\text{NLL}}(p_t, p_s) = -\log p_s(\mathbf{y}_t^u), \quad (10)$$

where \mathbf{y}_t^u is the mode predicted by the teacher and can be computed by SVD of \mathbf{A}_t^u (see Section 3.2).

Relationship between L^{NLL} and L^{CE} Here we intend to make connection between L^{NLL} and L^{CE} . We find that L^{CE} becomes L^{NLL} when we decreases the dispersion of the distribution p_t to a Dirac distribution $\delta(\mathbf{R}; \mathbf{y}_t^u)$ with its mode located at \mathbf{y}_t^u . We give a brief proof as below:

$$\begin{aligned} L^{\text{CE}}(\text{Dirac}(p_t), p_s) &= H(\delta(\mathbf{y}_t^u), p_s) \\ &= - \int_{SO(3)} \delta(\mathbf{y}_t^u) \log p_s d\mathbf{R} \\ &= -\log p_s(\mathbf{y}_t^u) = L^{\text{NLL}}(p_t, p_s). \end{aligned}$$

This exactly resembles the label sharpening technique used in semi-supervised classification [5,43], where the teacher’s output is either sharpened or turned into a hard label. To be specific, when we turn a predicted distribution $\mathcal{MF}(\mathbf{A}_t^u)$ into a hard label \mathbf{y}_t^u , L^{CE} becomes L^{NLL} . We use L^{CE} in the experiments and investigate the different behavior of these two losses in Section 4.4.

3.5. Training Protocol

Our training is composed of two stages: a pre-training stage, where we train our rotation regressor on the labeled data, followed by an SSL stage where both the labeled and the unlabeled data are utilized. Our matrix Fisher-based rotation regressor is fed with an RGB image \mathbf{x} and outputs a 3×3 matrix \mathbf{A} as the predicted parameter of the matrix Fisher distribution. We take the mode of the distribution as the predicted value.

Pre-training We start with a supervised training procedure on the labeled set with the supervised loss as Eq. 6. We clone the rotation regressor to obtain a pair of teacher and student networks with the same initialization, once converged.

Semi-supervised training In SSL stage, we utilize both the labeled data and the unlabeled data. A training batch contains a mixture of $\{\mathbf{x}_i^l\}_{i=1}^{B_l}$ labeled samples and $\{\mathbf{x}_i^u\}_{i=1}^{B_u}$ unlabeled samples. The loss function is composed of the supervised loss applied to the labeled samples and the unsupervised loss for the unlabeled samples

$$L = L_l(\mathbf{x}^l, \mathbf{y}^l) + \lambda_u L_u(\mathbf{x}^u) \quad (11)$$

where L_l is computed as Eq. 6, L_u is as Eq. 7, and λ_u is the unsupervised loss weight.

In this stage, We adopt asymmetric augmentation and an exponential-moving-average teacher as stated in Sec. 3.1.

4. Experiment

4.1. Datasets

ModelNet10-SO(3) [30] is created by rendering 3D models of ModelNet-10 [51] that are rotated by uniformly sampled random rotations in $\text{SO}(3)$. Following [9,35], we focus on the `chair` and `sofa` category which exhibit the least rotational symmetries in the dataset. In the experiments, we set the ratio of labeled data as 5% and 10% of the training set.

Pascal3D+ [52] contains real images from Pascal VOC and ImageNet of 12 rigid object classes. Following NVSM [46], we evaluate 6 vehicle categories (`aeroplane`, `bicycle`, `boat`, `bus`, `car`, `motorbike`) which have relatively evenly distributed poses in azimuth angles, and set the number of labeled images as 7, 20 and 50 for each category respectively. We share the same selected 7 images as NVSM such that they are spread around the pose space.

We follow the original train-test split and further divide the training split into the labeled set with ground truth and the unlabeled set without ground truth.

4.2. Evaluation setup

Baselines To the best of our knowledge, we are the first to tackle semi-supervised rotation regression in this setting, hence the comparisons are made with self-made baselines. **Supervised-L1** uses a normal regressor and only trains on the labeled set with L1 loss with the 9D-SVD [29] rotation representation, while **Supervised-Fisher** uses our matrix Fisher regressor and also only go through the pretraining stage. As an SSL baseline, **SSL-L1-Consistency** refers to adopting FixMatch into the task with the EMA teacher and asymmetric data augmentation preserved, but only applying L1 loss as the consistency supervision between the student and teacher predictions without filtering, due to lacking the confidence measure. Here, for non-Fisher regressors, we choose L1 instead of L2 loss, as [10] points out that L1 outperforms L2 for rotation regression.

We find the most relevant work to ours is NVSM [46], which, though not regression-based, tackles the same task as ours and leverages a render-and-compare scheme through distance-based rotation retrieval. We borrow NVSM and their developed baselines as our compared baselines, including two supervised rotation estimation works (**StarMap** [59] and **NeMo** [45]) and two standard classification networks (**Res50-Gene** and **Res50-Spec**), adapted into semi-supervised learning, respectively. Due to the unavailability of the training code, we exactly follow the experiment settings of NVSM and evaluate on Pascal3D+ dataset. See Appendix Section A for more details.

Evaluation metrics We evaluate the experiments by the mean error, the median error (in degrees) and the accuracy within 30° between the prediction and the ground truth.

4.3. Results

Result comparison Table 1 shows the results of our method compared with baselines on ModelNet10-SO(3) under different labeled data ratios. We can see that the results of supervised learning with the labeled data only perform similarly, regardless of using a normal or a Fisher regressor. Since these models are in fact the pre-trained models for the SSL methods in the SSL stage, their similar performance sets a common basis for a fair comparison in the SSL stage. For methods that undergo a second SSL stage, our proposed **FisherMatch** method consistently outperforms the baseline SSL method **SSL-L1-Consistency**, which demonstrates the importance of performing pseudo label filtering.

The experiment results on Pascal3D+ dataset are shown in Table 2. The results illustrate that, with the effective teacher-student mutual learning framework as well as the

Table 1. Comparing our proposed FisherMatch with the baselines on ModelNet10-SO(3) under different ratios of labeled data.

Category	Method	5%		10%	
		Mean↓	Med.↓	Mean↓	Med.↓
Sofa	Sup.-L1 [29]	44.64	11.42	32.65	9.03
	Sup.-Fisher [35]	45.19	13.16	32.92	8.83
	SSL-L1-Consist.	36.86	8.65	25.94	6.81
	SSL-FisherMatch	32.02	7.78	21.29	5.25
	Full Sup.	18.62	5.77	18.62	5.77
Chair	Sup.-L1 [29]	40.41	16.09	29.02	10.64
	Sup.-Fisher [35]	39.34	16.79	28.58	10.84
	SSL-L1-Consist.	31.20	11.29	23.59	8.10
	SSL-FisherMatch	26.69	9.42	20.06	7.44
	Full Sup.	17.38	6.78	17.38	6.78

Table 2. Comparing our proposed FisherMatch with the baselines on the 6 categories of Pascal3D+ dataset with few annotations (7, 20, 50 images). The results are averaged on 6 categories.

Method	7		20		50	
	Med.↓	Acc _{30°} ↑	Med.↓	Acc _{30°} ↑	Med.↓	Acc _{30°} ↑
Res50-Gene	39.1	36.1	26.3	45.2	20.2	54.6
Res50-Spec	46.5	29.6	29.4	42.8	23.0	50.4
StarMap [59]	49.6	30.7	46.4	35.6	27.9	53.8
NeMo [45]	60.0	38.4	33.3	51.7	22.1	69.3
NVSM [46]	37.5	53.8	28.7	61.7	24.2	65.6
FisherMatch	28.3	56.8	23.8	63.6	16.1	75.7
Full Sup.	8.1	89.6	8.1	89.6	8.1	89.6



Figure 2. The visualization of the training process of SSL-FisherMatch on ModelNet10-SO(3) Sofa dataset with 5% labeled data. The four plots, from left to right and from top to bottom, show the mean errors of the predictions, the pseudo label coverage, the pseudo label quality represented by the mean errors of the pseudo labels, and the mean errors between the student model and the corresponding pseudo labels, in the process of training. All the errors are measured in degrees.

entropy-based pseudo label filtering scheme, our algorithm significantly outperforms the state-of-the-art baselines under all different numbers of labeled images.

Training process analysis Here we show how our SSL method works during the training. In Fig. 2, the upper left plot shows that the performance of the unlabeled data

increases together with the test data, which indicates the increasing quality of the teacher predictions. We can also note that the performance on the unlabeled data is slightly better than that of the test data, which is sometimes referred to as *transductive semi-supervised learning*.

We also show the changes over the training process of the pseudo label coverage, the pseudo label quality, and the error between the student predictions and the corresponding pseudo labels, respectively. Here, we refer to *pseudo labels* as the teacher predictions that pass the entropy threshold. The pseudo label coverage means the percentage of teacher predictions that pass the confidence threshold. The pseudo label quality simply means the error of the pseudo labels to the ground truth.

As shown in the curves, as the SSL goes on, the improving model leads to more confident predictions indicated by the decreasing entropy and increasing pseudo label coverage, which in return fuels the learning process. The coverage of pseudo labels increases by a large margin from 40% to the final 70%, while the pseudo label quality still keeps stable with a shaking around 2.5°. This indicates that entropy always acts as a good indicator of performance during the whole process. The error of the student model to the pseudo labels keeps decreasing, which further proves the effectiveness of our unsupervised loss.

4.4. Ablation Study

Effect of Different Unsupervised Loss and Entropy Threshold Here, we analyze the performance of our Fish-

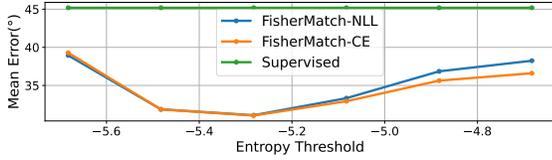


Figure 3. **The performance of FisherMatch with CE or NLL unsupervised losses with different entropy thresholds.** The experiments are done on ModelNet10-SO(3) Sofa dataset with 5% labeled data.

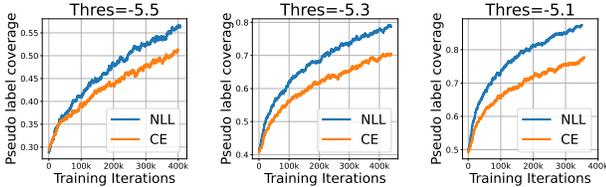


Figure 4. **Comparison of the pseudo label coverage over training process with CE or NLL unsupervised losses and entropy thresholds.** The experiments are done on ModelNet10-SO(3) Sofa dataset with 5% labeled data.

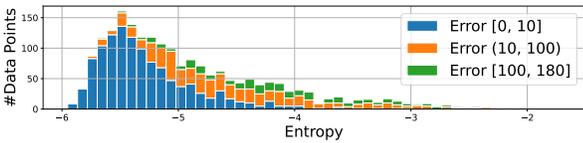


Figure 5. **Visualization of the indication ability of the distribution entropy wrt. the performance.** The horizontal axis is the distribution entropy and the vertical axis is the number of data points, color coded by the errors (in degrees). The experiments are done on ModelNet10-SO(3) Sofa dataset with 10% labeled data.

erMatch with different unsupervised losses, L^{CE} (Eq. 7) and L^{NLL} (Eq. 10), and how they are dependent on the entropy threshold τ by sweeping the parameter τ . Shown in Fig. 3 and 4, the CE loss performs slightly better with a more tolerant threshold, while the NLL loss encourages a higher confidence of the network. The results verify that the NLL loss is a sharpened version of CE loss, where all the pseudo labels passing the threshold are seen as *absolute confident* regardless of the actually predicted uncertainty. This behavior results in a more- but maybe over- confident network, especially with a tolerant entropy threshold. On the other hand, since pseudo labels already exhibit much confidence as they pass the threshold, further sharpening does not lead to additional performance gains. Thus we believe CE loss is a better choice in our task with broader compatibility.

Indication Ability of Distribution Entropy To clearly exhibit the indication ability of the distribution entropy wrt. the performance, we plot the relationship between the error of the prediction and the corresponding distribution entropy on test set in Fig. 5. The figure shows that the entropy effectively captures the prediction error, even under a low

Table 3. **Semi-supervised learning experiment based on the Bingham distribution** on ModelNet10-SO(3) Sofa dataset with 10% labeled data.

Method	Mean↓	Med.↓
Sup.-Bingham	39.61	12.68
Sup.-Fisher	32.92	8.83
SSL-BinghamMatch	27.01	6.77
SSL-FisherMatch	21.29	5.25

labeled data ratio.

Comparison with Bingham-based Regressor Our designed algorithm is agnostic to the choice of the rotation representation as well as the distribution model. We further test our framework based on the Bingham distribution and report the results in Table 3.

As shown in the table, the Bingham-based framework is also able to utilize the unlabeled data and significantly improve the performance of rotation estimation. However, for both its supervised and semi-supervised version, its rotation errors are in general larger than those of matrix Fisher-based framework, since its rotation representation, quaternion, is not a continuous rotation representation, as pointed in [60], thus leading to inferior performance. See Appendix Section A for detailed settings for SSL-BinghamMatch.

5. Conclusion and Limitations

In this paper, we tackle the problem of semi-supervised rotation regression from single RGB images in a general way. Without requiring any domain-specific knowledge or paired images, we leverage the teacher-student mutual learning framework and propose an entropy-based pseudo label filtering strategy based on the probabilistic modeling of SO(3). Our experiments demonstrate the effectiveness and advantage of our method on both ModelNet10-SO(3) and Pascal3D+ datasets.

The performance of our method may degrade when both the numbers of labeled and unlabeled data are not sufficient. In this case, the uncertainty predicted by our network can be under-estimated due to over-fitting in the small labeled data, leading to reduced effectiveness in the pseudo label filtering and thus the mutual learning.

Acknowledgements

We thank the anonymous reviewers for the insightful feedback. We would like to credit Jiangran Lv from DUT for the fruitful discussions and valuable help in experiments and Yang Wang from PKU for the help in the derivation of maths. This work is supported in part by grants from the Joint NSFC-ISF Research Grant (62161146002).

References

- [1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7822–7831, 2021. [15](#)
- [2] Arthur Asuncion and David Newman. Uci machine learning repository, 2007. [3](#)
- [3] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014. [3](#)
- [4] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. [3](#)
- [5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019. [1](#), [3](#), [6](#)
- [6] Christopher Bingham. An antipodally symmetric distribution on the sphere. *The Annals of Statistics*, pages 1201–1225, 1974. [12](#)
- [7] Michel Breyer, Jen Jen Chung, Lionel Ott, Siegwart Roland, and Nieto Juan. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. In *Conference on Robot Learning*, 2020. [1](#)
- [8] Mai Bui, Tolga Birdal, Haowen Deng, Shadi Albarqouni, Leonidas Guibas, Slobodan Ilic, and Nassir Navab. 6d camera relocalization in ambiguous scenes via continuous multimodal inference. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 139–157. Springer, 2020. [1](#)
- [9] Jiayi Chen, Yingda Yin, Tolga Birdal, Baoquan Chen, Leonidas Guibas, and He Wang. Projective manifold gradient layer for deep rotation regression. *arXiv preprint arXiv:2110.11657*, 2021. [6](#), [11](#)
- [10] Haowen Deng, Mai Bui, Nassir Navab, Leonidas Guibas, Slobodan Ilic, and Tolga Birdal. Deep bingham networks: Dealing with uncertainty and ambiguity in pose estimation. *arXiv preprint arXiv:2012.11002*, 2020. [2](#), [3](#), [5](#), [6](#), [11](#), [15](#)
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [11](#)
- [12] Thomas D Downs. Orientation statistics. *Biometrika*, 59(3):665–676, 1972. [2](#)
- [13] Jared Marshall Glover. *The quaternion Bingham distribution, 3D object detection, and dynamic manipulation*. PhD thesis, Massachusetts Institute of Technology, 2014. [5](#), [12](#)
- [14] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1759–1769, 2020. [1](#)
- [15] Chengyue Gong, Dilin Wang, and Qiang Liu. Alphamatch: Improving consistency for semi-supervised learning with alpha-divergence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13683–13692, 2021. [1](#)
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [11](#)
- [17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. [11](#)
- [18] Zijian Hu, Zhengyu Yang, Xuefeng Hu, and Ram Nevatia. Simple: Similar pseudo label exploitation for semi-supervised classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15099–15108, 2021. [3](#)
- [19] Neal Jean, Sang Michael Xie, and Stefano Ermon. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. *Advances in Neural Information Processing Systems*, 31, 2018. [3](#)
- [20] Zhenyu Jiang, Yifeng Zhu, Maxwell Svetlik, Kuan Fang, and Yuke Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. *Robotics: science and systems*, 2021. [1](#)
- [21] Peter E Jupp and Kanti V Mardia. Maximum likelihood estimators for the matrix von mises-fisher and bingham distributions. *The Annals of Statistics*, 7(3):599–606, 1979. [2](#)
- [22] Atul Kanaujia, Cristian Sminchisescu, and Dimitris Metaxas. Semi-supervised hierarchical models for 3d human pose reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. [1](#)
- [23] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6728–6736, 2019. [3](#)
- [24] CG Khatri and Kanti V Mardia. The von mises–fisher matrix distribution in orientation statistics. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):95–106, 1977. [2](#), [4](#), [13](#)
- [25] Plamen Koev and Alan Edelman. The efficient evaluation of the hypergeometric function of a matrix argument. *Mathematics of Computation*, 75(254):833–846, 2006. [15](#)
- [26] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. [1](#), [3](#)
- [27] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013. [3](#)
- [28] Taeyoung Lee. Bayesian attitude estimation with the matrix fisher distribution on so(3). *IEEE Transactions on Automatic Control*, 63(10):3377–3392, 2018. [14](#), [15](#), [16](#)
- [29] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snaveley, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameetesh

- Makadia. An analysis of svd for deep rotation estimation. *Advances in Neural Information Processing Systems*, 33:22554–22565, 2020. [2](#), [6](#), [7](#), [11](#)
- [30] Shuai Liao, Efstratios Gavves, and Cees GM Snoek. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9767, 2019. [6](#)
- [31] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14687–14697, 2021. [1](#)
- [32] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. [1](#), [4](#)
- [33] Octave Mariotti and Hakan Bilen. Semi-supervised viewpoint estimation with geometry-aware conditional generation. In *European Conference on Computer Vision*, pages 631–647. Springer, 2020. [3](#), [15](#)
- [34] Octave Mariotti, Oisín Mac Aodha, and Hakan Bilen. Viewnet: Unsupervised viewpoint estimation from conditional generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10418–10428, 2021. [1](#), [3](#)
- [35] David Mohlin, Josephine Sullivan, and Gérald Bianchi. Probabilistic orientation estimation with matrix fisher distributions. *Advances in Neural Information Processing Systems*, 33:4884–4893, 2020. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [14](#), [15](#), [16](#)
- [36] Kieran Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, and Ameesh Makadia. Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold. *arXiv preprint arXiv:2106.05965*, 2021. [2](#), [16](#)
- [37] Siva Karthik Mustikovela, Varun Jampani, Shalini De Mello, Sifei Liu, Umar Iqbal, Carsten Rother, and Jan Kautz. Self-supervised viewpoint learning from image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3971–3981, 2020. [3](#)
- [38] Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7241–7250, 2021. [3](#)
- [39] Dario Pavlo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. [1](#)
- [40] MJ Prentice. On invariant tests of uniformity for directions and orientations. *The Annals of Statistics*, pages 169–176, 1978. [2](#)
- [41] Michael J Prentice. Orientation statistics without parametric assumptions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(2):214–222, 1986. [4](#), [13](#)
- [42] Sergey Prokudin, Peter Gehler, and Sebastian Nowozin. Deep directional statistics: Pose estimation with uncertainty quantification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 534–551, 2018. [2](#)
- [43] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020. [1](#), [3](#), [4](#), [6](#)
- [44] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. [1](#), [3](#)
- [45] Angtian Wang, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. *arXiv preprint arXiv:2101.12378*, 2021. [6](#), [7](#), [11](#)
- [46] Angtian Wang, Shengxiao Mei, Alan L Yuille, and Adam Kortylewski. Neural view synthesis and matching for semi-supervised few-shot learning of 3d pose. *Advances in Neural Information Processing Systems*, 34, 2021. [3](#), [6](#), [7](#), [11](#)
- [47] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14615–14624, 2021. [1](#), [2](#), [4](#), [15](#)
- [48] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J. Guibas. Capra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 13209–13218, October 2021. [1](#)
- [49] Wikipedia contributors. Haar measure — Wikipedia, the free encyclopedia, 2022. [13](#), [14](#)
- [50] Wikipedia contributors. Lebesgue measure — Wikipedia, the free encyclopedia, 2022. [13](#)
- [51] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. [6](#)
- [52] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014. [1](#), [6](#)
- [53] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. [1](#)
- [54] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020. [3](#)
- [55] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR, 2021. [3](#)

- [56] Anna Yershova, Swati Jain, Steven M Lavalley, and Julie C Mitchell. Generating uniform incremental grids on so (3) using the hopf fibration. *The International journal of robotics research*, 29(7):801–812, 2010. **2, 16**
- [57] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020. **15**
- [58] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34, 2021. **3**
- [59] Xingyi Zhou, Arjun Karapur, Linjie Luo, and Qixing Huang. StarMap for category-agnostic keypoint and viewpoint estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018. **6, 7, 11**
- [60] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. **1, 2, 4, 8**
- [61] Zhi-Hua Zhou, Ming Li, et al. Semi-supervised regression with co-training. In *IJCAI*, volume 5, pages 908–913, 2005. **3**

A. Implementation and Experiment Details

A.1. Baselines in NVSM

In the main paper, we compare our algorithm with NVSM [46] and their developed baselines, *i.e.*, **StarMap**, **NeMo**, **Res50-Gene** and **Res50-Spec**. We briefly introduce these methods in this section, and more details can be found in [46].

StarMap [59] and NeMo [45] are two state-of-the-art supervised approaches for 3D pose estimation. For NeMo, the same single mesh cuboid is used as NVSM does. In addition, two baselines that formulate the object pose estimation problem as a classification task are adopted. To be specific, Res50-Gene formulates the pose estimation task for all categories as one single classification task, whereas Res50-Spec learns one classifier per category.

All baselines are evaluated using a semi-supervised protocol in a common pseudo labeling strategy. Specifically, all baselines are first trained on the annotated images and use the pretrained models to label the unlabeled data by pseudo labels. The final models are trained on both the annotated data and the pseudo-labeled data.

A.2. Experiment Settings of BinghamMatch

In Table 3 of the main paper, we experiment our algorithm based on the Bingham distribution $\mathcal{B}(\mathbf{M}, \mathbf{Z})$, namely BinghamMatch. We use the same experiment settings as FisherMatch, except that we choose unit quaternion as our

rotation representation and use Bingham distribution for building the probabilistic rotation model. The rotation regressor outputs the parameters of the Bingham distribution. Specifically, following [10], the regressor outputs a 7-d vector $(\mathbf{o}_1, \mathbf{o}_2)$ where the first 4-d vector \mathbf{o}_1 are first normalized and used to construct the parameter \mathbf{M} via *Birdal Strategy*

$$\mathbf{M}(\mathbf{o}_1) \triangleq \begin{bmatrix} o_{11} & -o_{12} & -o_{13} & o_{14} \\ o_{12} & o_{11} & o_{14} & o_{13} \\ o_{13} & -o_{14} & o_{11} & -o_{12} \\ o_{14} & o_{13} & -o_{12} & -o_{11} \end{bmatrix}$$

and the last 3-d vector \mathbf{o}_2 are applied by softplus activation and accumulation sum to construct the parameter \mathbf{Z} , with

$$\begin{aligned} z_1 &= -\phi(o_{21}) \\ z_2 &= -\phi(o_{21}) - \phi(o_{22}) \\ z_3 &= -\phi(o_{21}) - \phi(o_{22}) - \phi(o_{23}) \end{aligned}$$

where $\phi(\cdot)$ is the softplus activation.

A.3. Implementation Details

We run all the experiments with the unsupervised loss weight λ_u as 1. In the pre-training stage, we train with the batch size of 32, and for the SSL stage, a training batch is composed of 32 labeled samples and 128 unlabeled samples. Both the weak and strong augmentations consist of random padding, cropping, resizing and color jittering (for real-world images) operations with different strengths. On ModelNet10-SO(3) dataset, we use MobileNet-V2 [17] architecture following [9, 29]. We use the Adam optimizer with the learning rate as 1e-4 without decaying. The entropy threshold τ is set as around -5.3. On Pascal3D+ dataset, we follow NVSM [46] to use ResNet [16] architecture pre-trained on ImageNet [11] dataset. We use the Adam optimizer with the learning rate as 1e-4 in pre-training stage and 1e-5 in the Semi-supervised training stage, without decaying. Due to the extremely small amount of data, we find a large variation among experiments of different categories and #labeled images on Pascal3D+ dataset, thus choose different confidence thresholds in the SSL stage.

B. Review of Bingham Distribution and Matrix Fisher Distribution

B.1. Unit Quaternion and Rotation Matrix

Unit quaternion and rotation matrix are two commonly used representations for rotation elements from SO(3). Unit quaternion $\mathbf{q} \in \mathcal{S}^3$ is a double-covered representation of SO(3), where \mathbf{q} and $-\mathbf{q}$ represent the same rotation. Rotation $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ satisfies $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ and $\det(\mathbf{R}) = +1$. For a quaternion $\mathbf{q} = [w, x, y, z]$, we use the standard transform function γ to compute its corresponding rotation matrix:

$$\gamma(\mathbf{q}) = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 \end{bmatrix}$$

The inverse transform γ^{-1} is

$$\gamma^{-1}(\mathbf{R}) = \begin{bmatrix} \sqrt{1 + \mathbf{R}_{00} + \mathbf{R}_{11} + \mathbf{R}_{22}}/2 \\ (\mathbf{R}_{21} - \mathbf{R}_{12})/2\sqrt{1 + \mathbf{R}_{00} + \mathbf{R}_{11} + \mathbf{R}_{22}} \\ (\mathbf{R}_{02} - \mathbf{R}_{20})/2\sqrt{1 + \mathbf{R}_{00} + \mathbf{R}_{11} + \mathbf{R}_{22}} \\ (\mathbf{R}_{10} - \mathbf{R}_{01})/2\sqrt{1 + \mathbf{R}_{00} + \mathbf{R}_{11} + \mathbf{R}_{22}} \end{bmatrix}$$

Note that we here only cover one hemisphere of \mathcal{S}^3 .

B.2. Bingham Distribution

Bingham distribution [6, 13] is an antipodally symmetric distribution. Its probability density function $\mathcal{B} : \mathcal{S}^{d-1} \rightarrow \mathcal{R}$ is defined as

$$p_B(\mathbf{q}) = \mathcal{B}(\mathbf{q}; \mathbf{M}, \mathbf{Z}) = \frac{1}{F(\mathbf{Z})} \exp(\mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q}) \quad (12)$$

where $\mathbf{M} \in \text{O}(4)$ is a 4×4 orthogonal matrix and $\mathbf{Z} = \text{diag}(0, z_1, z_2, z_3)$ is a 4×4 diagonal matrix with $0 \geq z_1 \geq z_2 \geq z_3$. The first column of parameter \mathbf{M} indicates the mode and the remaining columns describe the orientation of dispersion while the corresponding z_i , ($i \in 1, 2, 3$) describe the strength of the dispersion. $F(\mathbf{Z})$ is the normalizing constant.

Proposition 1. *Given $f \sim \mathcal{B}(\mathbf{M}, \mathbf{Z})$, the entropy of Bingham distribution is computed as*

$$H_B(f) = \log F - \mathbf{Z} \frac{\nabla F}{F}. \quad (13)$$

Proof. Denote $C = \mathbf{M} \mathbf{Z} \mathbf{M}^T$

$$\begin{aligned} H_B(f) &= - \oint_{\mathbf{q} \in \mathcal{S}^3} f(\mathbf{q}) \log f(\mathbf{q}) d\mathbf{q} \\ &= - \oint_{\mathbf{q} \in \mathcal{S}^3} \frac{1}{F} \exp(\mathbf{q}^T C \mathbf{q}) (\mathbf{q}^T C \mathbf{q} - \log F) d\mathbf{q} \\ &= \log F - \frac{1}{F} \oint_{\mathbf{q} \in \mathcal{S}^3} \mathbf{q}^T C \mathbf{q} \exp(\mathbf{q}^T C \mathbf{q}). \end{aligned}$$

Writing f in standard form, and denoting the hyperspherical integral by $g(\mathbf{Z})$,

$$g(\mathbf{Z}) = \oint_{\mathbf{q} \in \mathcal{S}^3} \mathbf{q}^T C \mathbf{q} \exp(\mathbf{q}^T C \mathbf{q}) d\mathbf{q},$$

Then

$$\begin{aligned} g(\mathbf{Z}) &= \oint_{\mathbf{q} \in \mathcal{S}^3} \sum_{i=1}^4 z_i (\mathbf{v}_i^T \mathbf{q})^2 \exp\left(\sum_{j=1}^4 z_j (\mathbf{v}_j^T \mathbf{q})^2\right) d\mathbf{q} \\ &= \sum_{i=1}^4 z_i \frac{\partial F}{\partial z_i} = \mathbf{Z} \cdot \nabla F. \end{aligned}$$

Thus, the entropy is $\log F - \mathbf{Z} \frac{\nabla F}{F}$ \square

Proposition 2. *Given $f \sim \mathcal{B}(\mathbf{M}_f, \mathbf{Z}_f)$ and $g \sim \mathcal{B}(\mathbf{M}_g, \mathbf{Z}_g)$, the cross entropy between Bingham distributions (f to g) is computed as*

$$H_B(f, g) = \log F_g - \sum_{i=1}^4 z_{gi} \left(b_i^2 + \sum_{j=1}^4 (a_{ij}^2 - b_i^2) \frac{1}{F_f} \frac{\partial F_f}{\partial z_{fj}} \right). \quad (14)$$

where a_{ij} is the entries of $\hat{\mathbf{A}} = \mathbf{M}_f^T \mathbf{M}_g$ and b_i is the entries of $\mathbf{b} = \boldsymbol{\mu}_f^T \mathbf{M}_g$ ($\boldsymbol{\mu}_f$ is the mode of distribution f).

Proof.

$$\begin{aligned} H_B(f, g) &= - \oint_{\mathbf{q} \in \mathcal{S}^3} f(\mathbf{q}) \log g(\mathbf{q}) d\mathbf{q} \\ &= - \oint_{\mathbf{q} \in \mathcal{S}^3} f(\mathbf{q}) \left(\sum_{i=1}^4 z_{gi} (\mathbf{v}_{gi}^T \mathbf{q}) - \log F_g \right) d\mathbf{q} \\ &= \log F_g - \sum_{i=1}^4 z_{gi} E_f [(\mathbf{v}_{gi}^T \mathbf{q})]. \end{aligned}$$

Since $\begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} = \begin{bmatrix} \mathbf{M}_f^T \\ \boldsymbol{\mu}_f^T \end{bmatrix} \mathbf{M}_g$ and $\begin{bmatrix} \mathbf{M}_f^T \\ \boldsymbol{\mu}_f^T \end{bmatrix}$ is orthogonal, $\mathbf{M}_g = [\mathbf{M}_f \boldsymbol{\mu}_f] \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix}$, so $\mathbf{v}_{gi} = \mathbf{M}_f \mathbf{a}_i + b_i \boldsymbol{\mu}_f$. Thus,

$$\begin{aligned} E_f [(\mathbf{v}_{gi}^T \mathbf{q})] &= E_f \left[\left((\mathbf{M}_f \mathbf{a}_i + b_i \boldsymbol{\mu}_f)^T \mathbf{q} \right)^2 \right] \\ &= b_i^2 E_f [(\boldsymbol{\mu}_f^T \mathbf{q})^2] + \sum_{j=1}^4 a_{ij}^2 E_f [(\mathbf{v}_{fj}^T \mathbf{q})^2] \end{aligned}$$

by linearity of expectation, and since all the odd projected moments are zero. Since

$$E_f [(\boldsymbol{\mu}_f^T \mathbf{q})^2] = 1 - \sum_{j=1}^4 E_f [(\mathbf{v}_{fj}^T \mathbf{q})^2]$$

and

$$E_f [(\mathbf{v}_{fj}^T \mathbf{q})^2] = \frac{1}{F_f} \frac{\partial F_f}{\partial z_{fj}},$$

then

$$H(f, g) = \log F_g - \sum_{i=1}^4 z_{gi} \left(b_i^2 + \sum_{j=1}^4 (a_{ij}^2 - b_i^2) \frac{1}{F_f} \frac{\partial F_f}{\partial z_{fj}} \right). \quad \square$$

B.3. Matrix Fisher Distribution

Matrix Fisher distribution [24,41] $\mathcal{MF}(\mathbf{R}; \mathbf{A})$ is a probability distribution over $\text{SO}(3)$ for rotation matrices, whose probability density function is in the form of

$$p_F(\mathbf{R}) = \mathcal{MF}(\mathbf{R}; \mathbf{A}) = \frac{1}{F(\mathbf{A})} \exp(\text{tr}(\mathbf{A}^T \mathbf{R})) \quad (15)$$

where parameter $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ is an arbitrary 3×3 matrix and $F(\mathbf{A})$ is the normalizing constant. The mode and dispersion of the distribution can be computed from the singular value decomposition of the parameter \mathbf{A} . Assume $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and the singular values are sorted in descending order, the mode of the distribution is computed as

$$\hat{\mathbf{R}} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{UV}) \end{bmatrix} \mathbf{V}^T$$

and the singular values $\mathbf{S} = \text{diag}(s_1, s_2, s_3)$ indicates the strength of concentration. The larger a singular value s_i is, the more concentrated the distribution is along the corresponding axis (the i -th column of mode $\hat{\mathbf{R}}$).

Entropy and Cross Entropy Given $f \sim \mathcal{MF}(\mathbf{A}_f)$ and $g \sim \mathcal{MF}(\mathbf{A}_g)$, we can start with the definition,

$$H_F(f) = - \int_{\mathbf{R} \in \text{SO}(3)} f(\mathbf{R}) \log f(\mathbf{R}) d\mathbf{R}$$

and

$$H_F(f, g) = - \int_{\mathbf{R} \in \text{SO}(3)} f(\mathbf{R}) \log g(\mathbf{R}) d\mathbf{R}.$$

However, note the equivalence of matrix Fisher distribution and Bingham distribution (see Section B.4), and doing integrals over \mathbb{S}^3 (with 4 dimensions and 1 constraint) is easier than that over $\text{SO}(3)$ (with 9 dimensions and 6 constraints), we first convert a matrix Fisher distribution to its equivalent Bingham distribution, and compute the properties via the formula of Bingham distribution.

Let p_F be the pdf of a matrix Fisher distribution, and p_B be the pdf of its equivalent Bingham distribution. Based on Eq. 19 and 29 in Section B.4, we have

$$\begin{aligned} H_F(p_F) &= - \int_{\mathbf{R} \in \text{SO}(3)} p_F \log p_F d\mathbf{R} \\ &= - \oint_{\mathbf{q} \in \mathbb{S}^3} 2\pi^2 p_B (\log(2\pi^2) + \log(p_B)) \frac{1}{2\pi^2} d\mathbf{q} \\ &= - \log(2\pi^2) \oint_{\mathbf{q} \in \mathbb{S}^3} p_B d\mathbf{q} - \oint_{\mathbf{q} \in \mathbb{S}^3} p_B \log p_B d\mathbf{q} \\ &= H_B(p_B) - \log(2\pi^2). \end{aligned} \quad (16)$$

And similarly,

$$H_F(f, g) = H_B(f, g) - \log(2\pi^2). \quad (17)$$

B.4. Equivalence of Bingham Distribution and Matrix Fisher Distribution

As discussed in [41], for a random rotation matrix variable \mathbf{R} , it follows a matrix Fisher distribution if and only if its corresponding unit quaternion $\mathbf{q} = \gamma^{-1}(\mathbf{R})$ (γ is defined in Section B.1) follows a Bingham distribution, i.e., the matrix Fisher distribution is a reparameterization of the Bingham distribution.

In this section, we derive the fact of the equivalence of Bingham distribution and matrix Fisher distribution and clarify the relationships between the various parameters.

In measure theory, the *Lebesgue measure* [50] assigns a measure to subsets of n -dimensional Euclidean space, and the *Haar measure* [49] assigns an ‘‘invariant volume’’ to subsets of locally compact topological groups, in our case, the Lie group $\text{SO}(3)$. We define $d\mathbf{q}$ based on Lebesgue measure and $d\mathbf{R}$ based on Haar measure.

Proposition 3. *The scaling factor from unit quaternions to rotation matrices is constant, and satisfies*

$$d\mathbf{R} = \frac{1}{2\pi^2} d\mathbf{q} \quad (19)$$

Proof. Define S as the Lebesgue measure on \mathcal{S}^3 and T as the Haar measure on $\text{SO}(3)$. Generally we can write

$$T(d\mathbf{R}) = \alpha(\mathbf{q}) S(d\mathbf{q})$$

where $\alpha(\mathbf{q})$ is the scaling factor from unit quaternions to rotation matrices, or specifically,

$$\begin{aligned} T(d\mathbf{R}_1) &= \alpha(\mathbf{q}_1) S(d\mathbf{q}_1) \\ T(d\mathbf{R}_2) &= \alpha(\mathbf{q}_2) S(d\mathbf{q}_2) \end{aligned} \quad (20)$$

Due to the invariance of measure S on \mathcal{S}^3 , we have

$$S(d\mathbf{q}_1) = S(d\mathbf{q}_2) \quad (21)$$

Define ν as the mapping from \mathcal{S}^3 to $\text{SO}(3)$, i.e., $d\mathbf{R} = \nu(d\mathbf{q})$. Define \mathbf{h} as an element in \mathcal{S}^3 satisfying

$$\mathbf{h} d\mathbf{q}_1 = d\mathbf{q}_2$$

we then induce $\hat{\mathbf{h}} = \nu \circ \mathbf{h} \circ \nu^{-1}$ which is an element in $\text{SO}(3)$, which thus satisfies

$$\hat{\mathbf{h}} \nu(d\mathbf{q}_1) = \nu(d\mathbf{q}_2)$$

Due to the invariance of measure T on $\text{SO}(3)$ [49], we have

$$T(\hat{\mathbf{h}} \nu(d\mathbf{q}_1)) = T(\nu(d\mathbf{q}_1)) = T(\nu(d\mathbf{q}_2))$$

i.e.,

$$T(d\mathbf{R}_1) = T(d\mathbf{R}_2) \quad (22)$$

$$\mathbf{A} = \mathbf{U}_1 \mathbf{S}' \mathbf{V}_1^T = \underbrace{\mathbf{U}_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U}_1) \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} s'_1 & 0 & 0 \\ 0 & s'_2 & 0 \\ 0 & 0 & \det(\mathbf{U}_1 \mathbf{V}_1) s'_3 \end{bmatrix}}_{\mathbf{S}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}_1) \end{bmatrix}}_{\mathbf{V}^T} \mathbf{V}_1^T = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (18)$$

Considering arbitrary $d\mathbf{q}_1$ and $d\mathbf{q}_2$, and based on Eq. 20, 21 and 22, we can derive that $\alpha(\mathbf{q})$ is a constant, i.e.,

$$d\mathbf{R} = \alpha d\mathbf{q}. \quad (23)$$

Known that the Haar measure is uniquely specified by adding the normalization condition [49], we have

$$\int_{\mathbf{R} \in \text{SO}(3)} d\mathbf{R} = 1$$

and based on the definition of unit quaternions,

$$\oint_{\mathbf{q} \in \mathcal{S}^3} d\mathbf{q} = |\mathcal{S}^3| = 2\pi^2$$

According to Eq. 23, we can derive that

$$d\mathbf{R} = \frac{1}{2\pi^2} d\mathbf{q}$$

as claimed. \square

Let \mathbf{I}_n be the n -dimensional identity matrix, and $\epsilon_i, i = 1, 2, \dots, n$ be the columns of \mathbf{I}_n . Let $\mathbf{E}_i = 2\epsilon_i \epsilon_i^T - \mathbf{I}_3, i = 1, 2, 3$ and $\mathbf{E}_4 = \mathbf{I}_3$. Define $Q(\mathbf{X})$ for a 3×3 rotation matrix as

$$4Q(\mathbf{X}) = 4\mathbf{x}\mathbf{x}^T - \mathbf{I}_4 \quad (24)$$

where $\mathbf{x} = \gamma^{-1}(\mathbf{X})$. Apply *proper* singular value decomposition [28, 35] to \mathbf{A} as Eq. 18

$$\mathbf{A} = \mathbf{U} \mathbf{S} \mathbf{V}^T$$

where \mathbf{U} and \mathbf{V} are guaranteed to be rotation matrices and \mathbf{S} contains the *proper* singular values with $s_1 \geq s_2 \geq |s_3|$. Define $T(\mathbf{A})$ for any real 3×3 matrix \mathbf{A} as

$$4T(\mathbf{A}) = \sum_{i=1}^4 z_i Q(\mathbf{U} \mathbf{E}_i \mathbf{V}). \quad (25)$$

Let z_1, z_2, z_3, z_4 denote the entries of \mathbf{Z} and $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4$ denote the columns of \mathbf{M} .

Proposition 4. *Suppose the parameters satisfy the following relationships*

$$\mathbf{Z} = 4T(\mathbf{S}) \quad (26)$$

$$\mathbf{m}_i = \gamma^{-1}(\mathbf{U} \mathbf{E}_i \mathbf{V}^T), i = 1, 2, 3, 4 \quad (27)$$

and the inputs

$$\mathbf{R} = \gamma(\mathbf{q}),$$

matrix Fisher distribution is equivalent to Bingham distribution with the relationship

$$\text{tr}(\mathbf{A} \mathbf{R}^T) = \mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q} \quad (28)$$

and

$$p_F(\mathbf{R}) = 2\pi^2 p_B(\mathbf{q}) \quad (29)$$

Proof. Assume $\mathbf{S} = \text{diag}(s_1, s_2, s_3)$ then we may write

$$4\mathbf{A} = \sum_{i=1}^4 z_i \mathbf{U} \mathbf{E}_i \mathbf{V}^T$$

uniquely, with

$$\begin{aligned} z_1 &= s_1 - s_2 - s_3 \\ z_2 &= s_2 - s_1 - s_3 \\ z_3 &= s_3 - s_1 - s_2 \\ z_4 &= -z_1 - z_2 - z_3. \end{aligned}$$

Also, since $4\mathbf{E}_i = 3\mathbf{E}_i - \sum_j \mathbf{E}_j, i \neq j$, Eq. 25 agrees with Eq. 24 on $\text{SO}(3)$. Assume $\gamma(\mathbf{m}_i) = \mathbf{U} \mathbf{E}_i \mathbf{V}^T, i = 1, 2, 3, 4$, then \mathbf{m}_i are mutually orthogonal, since $\text{tr}(\gamma(\mathbf{m}_i) \gamma(\mathbf{m}_j)^T) = -1$ if $i \neq j$. Hence we may write

$$4T(\mathbf{A}) = \mathbf{M} \mathbf{Z} \mathbf{M}^T$$

where $\mathbf{Z} = \text{diag}(z_1, z_2, z_3, z_4)$ has a zero trace and $\mathbf{M} = (\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4)$ in $\text{SO}(4)$. Note that

$$4T(\mathbf{S}) = \mathbf{Z}$$

and

$$4 \text{tr}(\mathbf{A} \mathbf{R}^T) = \sum_{i=1}^4 z_i \text{tr}(\mathbf{U} \mathbf{E}_i \mathbf{V}^T \mathbf{R}^T),$$

we have

$$\text{tr}(\mathbf{A} \mathbf{R}^T) = \mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q} \quad (30)$$

Due to the scaling factor from unit quaternions to rotation matrices is constant (See Prop. 3), matrix Fisher distribution is equivalent to Bingham distribution. Based on Eq. 30 and 19, and the conservation of the total probability, it can be shown that

$$p_F(\mathbf{R}) = 2\pi^2 p_B(\mathbf{q})$$

as claimed. \square

Note that the proposition can also be verified by the relationships between the normalization constant $F_B(\mathbf{Z})$ and $F_F(\mathbf{A})$. As discussed in [10, 28, 35], when \mathbf{Z} satisfies Eq. 26, the constant

$$\begin{aligned} F_B(\mathbf{Z}) &= \oint_{\mathbf{q} \in \mathbb{S}^3} \exp(\mathbf{q}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \mathbf{q}) \, d\mathbf{q} = |\mathbb{S}^3| {}_1F_1\left(\frac{1}{2}, 2, \mathbf{Z}\right) \\ &= 2\pi^2 {}_1F_1\left(\frac{1}{2}, 2, \mathbf{Z}\right) \end{aligned}$$

and

$$F_F(\mathbf{A}) = \int_{\mathbf{R} \in \text{SO}(3)} \exp(\text{tr}(\mathbf{A}^T \mathbf{R})) \, d\mathbf{R} = {}_1F_1\left(\frac{1}{2}, 2, \mathbf{Z}\right)$$

where ${}_1F_1(\cdot, \cdot, \cdot)$ is the generalized hypergeometric function [25] of a matrix argument. So

$$F_F(\mathbf{Z}) = \frac{1}{2\pi^2} F_F(\mathbf{A}).$$

Considering Eq. 30, we have

$$p_F(\mathbf{R}) = 2\pi^2 p_B(\mathbf{q})$$

B.5. Normalization Constant of Matrix Fisher Distribution

We follow [35] to compute the normalization constant. As pointed in [28], the normalizing constant of matrix Fisher distribution can be expressed as a one dimensional integral over Bessel functions as

$$\begin{aligned} c(S) &= \int_{-1}^1 \frac{1}{2} I_0 \left[\frac{1}{2} (s_i - s_j) (1 - u) \right] \\ &\quad \times I_0 \left[\frac{1}{2} (s_i + s_j) (1 + u) \right] \exp(s_k u) \, du \end{aligned}$$

and

$$\begin{aligned} \frac{\partial c(S)}{\partial s_i} &= \int_{-1}^1 \frac{1}{4} (1 - u) I_1 \left[\frac{1}{2} (s_i - s_j) (1 - u) \right] \\ &\quad \times I_0 \left[\frac{1}{2} (s_i + s_j) (1 + u) \right] \exp(s_k u) \\ &\quad + \frac{1}{4} (1 + u) I_0 \left[\frac{1}{2} (s_i - s_j) (1 - u) \right] \\ &\quad \times I_1 \left[\frac{1}{2} (s_i + s_j) (1 + u) \right] \exp(s_k u) \, du \end{aligned}$$

$$\begin{aligned} \frac{\partial c(S)}{\partial s_j} &= \int_{-1}^1 -\frac{1}{4} (1 - u) I_1 \left[\frac{1}{2} (s_i - s_j) (1 - u) \right] \\ &\quad \times I_0 \left[\frac{1}{2} (s_i + s_j) (1 + u) \right] \exp(s_k u) \\ &\quad + \frac{1}{4} (1 + u) I_0 \left[\frac{1}{2} (s_i - s_j) (1 - u) \right] \\ &\quad \times I_1 \left[\frac{1}{2} (s_i + s_j) (1 + u) \right] \exp(s_k u) \, du \end{aligned}$$

$$\begin{aligned} \frac{\partial c(S)}{\partial s_k} &= \int_{-1}^1 \frac{1}{2} I_0 \left[\frac{1}{2} (s_i - s_j) (1 - u) \right] \\ &\quad \times I_0 \left[\frac{1}{2} (s_i + s_j) (1 + u) \right] u \exp(s_k u) \, du \end{aligned}$$

for any $(i, j, k) \in \mathcal{I}$.

We approximate this integral using the trapezoid rule, where in experiments, 511 trapezoids are used. We use standard polynomials to approximate the Bessel function using Horner's method.

Please see Section 5 of [35]'s supplementary for more details.

C. More Experiment Results

C.1. Results on ModelNet10-SO(3) Dataset with 100% Labeled Data

Although out of the scope of semi-supervised learning, following [33, 47], we also report the results on 100% labeled data on ModelNet10-SO(3) dataset, where we simply make a copy of the full training data as unlabeled data and train our model. All the other settings are kept the same as Table 1 in the main paper.

As shown in Table 4, our proposed FisherMatch is able to further encourage a better performance with 100% labeled data compared with the supervised learning and consistently outperforms other baselines. The results further demonstrate the importance of filtering high-quality pseudo labels even with much training data. The improvements can be seen as a result of label smoothing [57].

C.2. Experiments and Results on Objectron Dataset

Dataset Objectron [1] is a newly-introduced dataset captured in the real world. The dataset contains a collection of short, object-centric video clips, as well as the corresponding camera poses, sparse point clouds, and manually annotated 3D bounding boxes for each object.

In this experiment, we mainly focus on the `bike` and `camera` categories which exhibit more rotational variations and less rotational symmetries in the dataset [1]. Since the real-world images are mostly captured from limited viewpoints, we found a smaller generalization gap between the train/test data. Thus, we choose a more challenging scenario to only adopt 1% labeled data to train the network. We adopt the official train-test split of the dataset, where we grab all the frames of the training videos and uniformly sample 10% frames from the test videos. We further divide the training split into the labeled set with ground truth and the unlabeled set without ground truth.

Data preprocessing To leverage this dataset for object pose regression, we need to obtain the paired data,

Table 4. Comparing our proposed FisherMatch with the baselines on ModelNet10-SO(3) dataset under 100% labeled data.

Category	Method	100%	
		Mean↓	Med.↓
Sofa	Sup.-L1	19.28	6.64
	Sup.-Fisher	18.62	5.77
	SSL-L1-Consist.	17.18	5.27
	SSL-FisherMatch	14.37	4.32
Chair	Sup.-L1	17.65	7.48
	Sup.-Fisher	17.38	6.78
	SSL-L1-Consist.	14.78	6.19
	SSL-FisherMatch	13.01	5.35

Table 5. Comparing our proposed FisherMatch with the baselines on Objectron dataset with 1% labeled data.

Category	Method	1%	
		Mean↓	Med.↓
Bike	Sup.-L1	53.6	21.2
	Sup.-Fisher	51.2	24.0
	SSL-L1-Consist.	38.0	14.3
	SSL-FisherMatch	36.0	13.8
	Full sup.	26.7	9.7
Camera	Sup.-L1	46.0	22.8
	Sup.-Fisher	39.0	18.7
	SSL-L1-Consist.	40.9	19.0
	SSL-FisherMatch	33.6	15.9
	Full sup.	24.4	9.5

i.e., object-centered images with their corresponding object poses. We thus first project the eight corners of 3D bounding box annotations onto the 2D image plane, fit a minimum 2D square bounding box covering all the projected corners, and finally crop the image with the fitted 2D bounding box. To avoid the naive cropping-resizing flaws pointed out in [35], we directly crop square images to meet the shape requirement of the network. We pad the images with a black background to cover the out-of-plane projected keypoints and images with more than 4 (out of 8) keypoints out of the image plane are discarded. To obtain the ground-truth object poses, we compute the rotation of the annotated 3D object bounding box wrt. the box with the same size in the canonical orientation.

Experiment settings The baselines, evaluation metrics and implementation details are the same as experiments on ModelNet10-SO(3) dataset.

Results The results are shown in Table 5. Our FisherMatch significantly increases the regression performance even with a really low labeled data ratio, further demonstrating the efficiency of our model.

D. Visualization of Matrix Fisher Distribution

Visualizing matrix Fisher distribution is non-trivial over SO(3). Following [28, 35], we visualize the probabilistic distribution function via color-coding on the sphere.

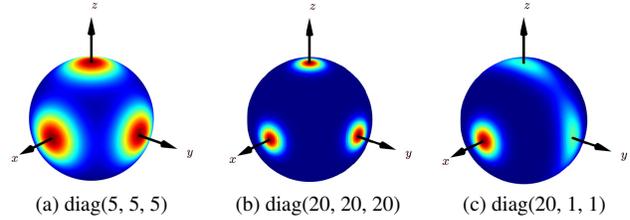


Figure 6. Visualization of the pdf of matrix Fisher distribution with jet color-coding. The captions below the plots indicate the parameter \mathbf{A} of the distribution.

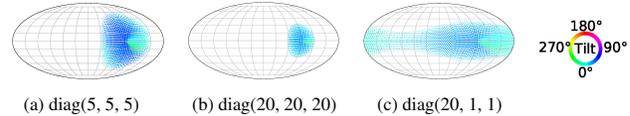


Figure 7. Visualization of the pdf of matrix Fisher distribution with the visualization method proposed in Implicit-PDF [36]. The captions below the plots indicate the parameter \mathbf{A} of the distribution.

Remember that for the parameter \mathbf{A} in matrix Fisher distribution, the singular values indicate the strength of concentration. The larger a singular value s_i is, the more concentrated the distribution is along the corresponding axis. Fig 6 shows three distributions with the same mode as the identity matrix, differing only in the strength of concentration. For both (a) and (b), the distributions of each axis are identical and circular, while the distribution in (b) is more concentrated than (a). In (c), the distribution is more concentrated in x -axis, and the distributions for the other two axes are elongated.

Implicit-PDF [36] proposes a new visualization method to display distributions over SO(3) by discretizing SO(3) with the help of Hopf fibration [56]. It projects a great circle of points on SO(3) to each point on the 2-sphere and uses the color wheel to indicate the location on the great circle. We re-draw Figure 6 with this visualization in Figure 7.