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

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

Abstract

In this supplementary material, we provide more details of the implementation and experiment settings in Section A, a brief review of Bingham distribution and matrix Fisher distribution as well as the derivation of important properties in Section B, more experiment results in Section C, and the distribution visualization method in Section D.

A. Implementation and Experiment Details

A.1. Baselines in NVSM

In the main paper, we compare our algorithm with NVSM [18] 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 [18].

StarMap [24] and NeMo [17] 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 [4], 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 [8] architecture following [3, 12]. 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 [18] to use ResNet [7] architecture pretrained on ImageNet [5] 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.

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

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 $\mathrm{SO}(3)$. Unit quaternion $\mathbf{q} \in \mathcal{S}^3$ is a double-covered representation of $\mathrm{SO}(3)$, where $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation. Rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ 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 [2, 6] is an antipodally symmetric distribution. Its probability density function $\mathcal{B} : \mathcal{S}^{d-1} \to \mathcal{R}$ is defined as

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

where $\mathbf{M} \in \mathrm{O}(4)$ is a 4×4 orthogonal matrix and $\mathbf{Z} = \mathrm{diag}(0, z_1, z_2, z_3)$ 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*

$$H_B(f) = \log F - \mathbf{Z}\frac{\nabla F}{F}. \quad (2)$$

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

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

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

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

Then

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

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

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*

$$H_B(f, g) = \log F_g - \sum_{i=1}^{4} z_{gi} \left(b_i^2 + \sum_{j=1}^{4} \left(a_{ij}^2 - b_i^2\right) \frac{1}{F_f} \frac{\partial F_f}{\partial z_{fj}}\right). \quad (3)$$

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

Proof.

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

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

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

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

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

and

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

then

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

□

B.3. Matrix Fisher Distribution

Matrix Fisher distribution [9, 16] $\mathcal{MF}(\mathbf{R};\mathbf{A})$ is a probability distribution over SO(3) for rotation matrices, whose probability density function is in the form of

$$p_F(\mathbf{R}) = \mathcal{MF}(\mathbf{R};\mathbf{A}) = \frac{1}{F(\mathbf{A})}\exp\left(\mathrm{tr}\left(\mathbf{A}^T\mathbf{R}\right)\right) \quad (4)$$

where parameter $\mathbf{A} \in \mathbb{R}^{3\times3}$ 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 $\mathbf{A} = \mathbf{USV}^T$ 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 $\mathbf{S} = \mathrm{diag}(s_1, s_2, s_3)$ 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,

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

and

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

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 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. 8 and 18 in Section B.4, we have

$$\begin{aligned} H_F(p_F) &= -\int_{\mathbf{R}\in\mathrm{SO}(3)} p_F\log p_F\mathrm{d}\mathbf{R}\\ &= -\oint_{\mathbf{q}\in\mathbb{S}^3} 2\pi^2 p_B\left(\log(2\pi^2) + \log(p_B)\right)\frac{1}{2\pi^2}\mathrm{d}\mathbf{q}\\ &= -\log(2\pi^2)\oint_{\mathbf{q}\in\mathbb{S}^3} p_B\mathrm{d}\mathbf{q} - \oint_{\mathbf{q}\in\mathbb{S}^3} p_B\log\mathrm{d}\mathbf{q}\\ &= H_B(p_B) - \log(2\pi^2). \end{aligned} \quad (5)$$

And similarly,

$$H_F(f,g) = H_B(f,g) - \log(2\pi^2). \quad (6)$$

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

As discussed in [16], 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* [21] assigns a measure to subsets of n-dimensional Euclidean space, and the *Haar measure* [20] assigns an "invariant volume" to subsets of locally compact topological groups, in our case, the Lie group $\mathrm{SO}(3)$. We define $\mathrm{d}\mathbf{q}$ based on Lebesgue measure and $\mathrm{d}\mathbf{R}$ based on Haar measure.

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

$$\mathrm{d}\mathbf{R} = \frac{1}{2\pi^2}\mathrm{d}\mathbf{q} \quad (8)$$

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

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

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

$$\begin{aligned} T(\mathrm{d}\mathbf{R_1}) &= \alpha(\mathbf{q_1})S(\mathrm{d}\mathbf{q_1})\\ T(\mathrm{d}\mathbf{R_2}) &= \alpha(\mathbf{q_2})S(\mathrm{d}\mathbf{q_2}) \end{aligned} \quad (9)$$

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

$$S(\mathrm{d}\mathbf{q_1}) = S(\mathrm{d}\mathbf{q_2}) \quad (10)$$

$$\mathbf{A}=\mathbf{U}_1\mathbf{S}'\mathbf{V}_1^T=\underbrace{\mathbf{U}_1\begin{bmatrix}1&0&0\\0&1&0\\0&0&\det(\mathbf{U}_1)\end{bmatrix}}_{\mathbf{U}}\underbrace{\begin{bmatrix}s_1'&0&0\\0&s_2'&0\\0&0&\det(\mathbf{U}_1\mathbf{V}_1)\,s_3'\end{bmatrix}}_{\mathbf{S}}\underbrace{\begin{bmatrix}1&0&0\\0&1&0\\0&0&\det(\mathbf{V}_1)\end{bmatrix}\mathbf{V}_1^T}_{\mathbf{V}^T}=\mathbf{USV}^T \tag{7}$$

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

$$\mathbf{h}\mathrm{d}\mathbf{q_1}=\mathrm{d}\mathbf{q_2}$$

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

$$\hat{\mathbf{h}}\nu\left(\mathrm{d}\mathbf{q_1}\right)=\nu\left(\mathrm{d}\mathbf{q_2}\right)$$

Due to the invariance of measure T on SO(3) [20], we have

$$T(\hat{\mathbf{h}}\nu\left(\mathrm{d}\mathbf{q_1}\right))=T\left(\nu\left(\mathrm{d}\mathbf{q_1}\right)\right)=T\left(\nu\left(\mathrm{d}\mathbf{q_2}\right)\right)$$

i.e.,

$$T\left(\mathrm{d}\mathbf{R_1}\right)=T\left(\mathrm{d}\mathbf{R_2}\right) \tag{11}$$

Considering arbitrary $\mathrm{d}\mathbf{q_1}$ and $\mathrm{d}\mathbf{q_2}$, and based on Eq. 9, 10 and 11, we can derive that $\alpha(\mathbf{q})$ is a constant, *i.e.*,

$$\mathrm{d}\mathbf{R}=\alpha\mathrm{d}\mathbf{q}. \tag{12}$$

Known that the Haar measure is uniquely specified by adding the normalization condition [20], we have

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

and based on the definition of unit quaternions,

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

According to Eq. 12, we can derive that

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

as claimed. □

Let $\mathbf{I}_n$ be the n-dimensional identity matrix, and $\epsilon_i, i=1,2,\ldots,n$ 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

$$4Q(\mathbf{X})=4\mathbf{x}\mathbf{x}^T-\mathbf{I}_4 \tag{13}$$

where $\mathbf{x}=\gamma^{-1}(\mathbf{X})$. Apply *proper* singular value decomposition [11, 14] to $\mathbf{A}$ as Eq. 7

$$\mathbf{A}=\mathbf{USV}^T$$

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

$$4T(\mathbf{A})=\sum_{i=1}^{4}z_iQ(\mathbf{UE_iV}). \tag{14}$$

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*

$$\mathbf{Z}=4T(\mathbf{S}) \tag{15}$$

$$\mathbf{m}_i=\gamma^{-1}(\mathbf{UE}_i\mathbf{V}^T), i=1,2,3,4 \tag{16}$$

and the inputs

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

matrix Fisher distribution is equivalent to Bingham distribution with the relationship

$$\mathrm{tr}(\mathbf{AR}^T)=\mathbf{q}^T\mathbf{MZM}^T\mathbf{q} \tag{17}$$

and

$$p_F(\mathbf{R})=2\pi^2p_B(\mathbf{q}) \tag{18}$$

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

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

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. 14 agrees with Eq. 13 on SO(3). Assmue $\gamma(\mathbf{m}_i)=\mathbf{UE}_i\mathbf{V}^T, i=1,2,3,4$, then $\mathbf{m}_i$ are mutually orthogonal, since $\mathrm{tr}\left(\gamma(\mathbf{m}_i)\gamma(\mathbf{m}_j)^T\right)=-1$ if $i\neq j$. Hence we may write

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

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

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

and

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

we have

$$\operatorname{tr}(\mathbf{A}\mathbf{R}^T) = \mathbf{q}^T\mathbf{M}\mathbf{Z}\mathbf{M}^T\mathbf{q} \tag{19}$$

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. 19 and 8, 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. □

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 [4, 11, 14], when $\mathbf{Z}$ satisfies Eq. 15, the constant

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

and

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

where ${}_1F_1(\cdot,\cdot,\cdot)$ is the generalized hypergeometric function [10] of a matrix argument. So

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

Considering Eq. 19, we have

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

B.5. Normalization Constant of Matrix Fisher Distribution

We follow [14] to compute the normalization constant. As pointed in [11], the normalizing constant of matrix Fisher distribution can be expressed as a one dimensional integral over Bessel functions as

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

and

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

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

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

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 [14]'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 [13, 19], 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 1, 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 [23].

Table 1. **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 2. **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

C.2. Experiments and Results on Objectron Dataset

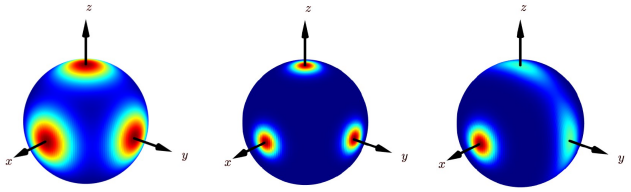
Dataset Objectron [1] is a newly-introduced dataset captured in the real world. The dataset contains a collection of short, object-centric video clips, as well as the corresponding camera poses, sparse point clouds, and manually annotated 3D bounding boxes for each object.

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, *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 [14], 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.

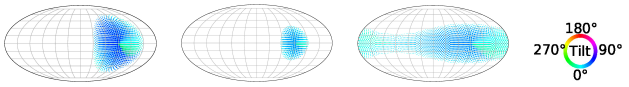
Experiment settings The baselines, evaluation metrics and implementation details are the same as experiments on ModelNet10-SO(3) dataset.

Results The results are shown in Table 2. Our FisherMatch significantly increases the regression performance even with a really low labeled data ratio, further demonstrating the efficiency of our model.



(a) diag(5, 5, 5) (b) diag(20, 20, 20) (c) diag(20, 1, 1)

Figure 1. **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.



(a) diag(5, 5, 5) (b) diag(20, 20, 20) (c) diag(20, 1, 1)

Figure 2. **Visualization of the pdf of matrix Fisher distribution** with the visualization method proposed in Implicit-PDF [15]. The captions below the plots indicate the parameter $\mathbf{A}$ of the distribution.

D. Visualization of Matrix Fisher Distribution

Visualizing matrix Fisher distribution is non-trivial over SO(3). Following [11, 14], we visualize the probabilistic distribution function via color-coding on the sphere.

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 1 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 [15] proposes a new visualization method to display distributions over $\mathrm{SO}(3)$ by discretizing $\mathrm{SO}(3)$ with the help of Hopf fibration [22]. It projects a great circle of points on $\mathrm{SO}(3)$ to each point on the 2-sphere and uses the color wheel to indicate the location on the great circle. We re-draw Figure 1 with this visualization in Figure 2.

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. 6

[2] Christopher Bingham. An antipodally symmetric distribution on the sphere. *The Annals of Statistics*, pages 1201–1225, 1974. 2

[3] 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. 1

[4] 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. 1, 5

[5] 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. 1

[6] Jared Marshall Glover. *The quaternion Bingham distribution, 3D object detection, and dynamic manipulation*. PhD thesis, Massachusetts Institute of Technology, 2014. 2

[7] 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. 1

[8] 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. 1

[9] 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. 3

[10] Plamen Koev and Alan Edelman. The efficient evaluation of the hypergeometric function of a matrix argument. *Mathematics of Computation*, 75(254):833–846, 2006. 5

[11] Taeyoung Lee. Bayesian attitude estimation with the matrix fisher distribution on so (3). *IEEE Transactions on Automatic Control*, 63(10):3377–3392, 2018. 4, 5, 6

[12] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. *Advances in Neural Information Processing Systems*, 33:22554–22565, 2020. 1

[13] 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. 5

[14] David Mohlin, Josephine Sullivan, and Gérald Bianchi. Probabilistic orientation estimation with matrix fisher distributions. *Advances in Neural Information Processing Systems*, 33:4884–4893, 2020. 4, 5, 6

[15] 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. 6, 7

[16] Michael J Prentice. Orientation statistics without parametric assumptions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(2):214–222, 1986. 3

[17] 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. 1

[18] Angtian Wang, Shenxiao 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. 1

[19] 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. 5

[20] Wikipedia contributors. Haar measure — Wikipedia, the free encyclopedia, 2022. 3, 4

[21] Wikipedia contributors. Lebesgue measure — Wikipedia, the free encyclopedia, 2022. 3

[22] Anna Yershova, Swati Jain, Steven M Lavalle, 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. 7

[23] 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. 5

[24] Xingyi Zhou, Arjun Karpur, 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. 1