$SE(3)$ Equivariant Ray Embeddings for Implicit Multi-View Depth Estimation

Yinshuang Xu
University of Pennsylvania
xuyin@seas.upenn.edu

Dian Chen
Toyota Research Institute
dian.chen@tri.global

Katherine Liu
Toyota Research Institute
katherine.liu@tri.global

Sergey Zakharov
Toyota Research Institute
sergey.zakharov@tri.global

Rares Ambrus
Toyota Research Institute
rares.ambrus@tri.global

Kostas Daniilidis
University of Pennsylvania
kostas@cis.upenn.edu

Vitor Guizilini
Toyota Research Institute
vitor.guizilini@tri.global

Abstract

Incorporating inductive bias by embedding geometric entities (such as rays) as input has proven successful in multi-view learning. However, the methods adopting this technique typically lack equivariance, which is crucial for effective 3D learning. Equivariance serves as a valuable inductive prior, aiding in the generation of robust multi-view features for 3D scene understanding. In this paper, we explore the application of equivariant multi-view learning to depth estimation, not only recognizing its significance for computer vision and robotics but also addressing the limitations of previous research. Most prior studies have either overlooked equivariance in this setting or achieved only approximate equivariance through data augmentation, which often leads to inconsistencies across different reference frames. To address this issue, we propose to embed $SE(3)$ equivariance into the Perceiver IO architecture. We employ Spherical Harmonics for positional encoding to ensure 3D rotation equivariance, and develop a specialized equivariant encoder and decoder within the Perceiver IO architecture. To validate our model, we applied it to the task of stereo depth estimation, achieving state of the art results on real-world datasets without explicit geometric constraints or extensive data augmentation.

1 Introduction

Equivariance is a valuable property in computer vision, leveraging various symmetries to reduce sample and model complexity while boosting generalization. It has seen broad application in fields such as 3D shape analysis [48, 52, 13], panoramic image prediction [10, 54, 19], and robotics [46, 25, 42, 2]. In particular, there is an increasing interest in equivariant scene representation from multiple viewpoints [43, 55], as the multi-view setting is a fundamental challenge in the field and equivariant representations are desirable for their robustness and efficiency.

Meanwhile, multi-view depth estimation has always been a core topic in computer vision. Previous works [27, 31, 26] leverage the explicit geometric constraint to construct the feature cost-volume for depth prediction. Recently, the paradigm of combining implicit representations with generalist

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

architectures has been widely adopted and gaining success. Inserting inductive bias via the embedding of geometric entities (rays) in the multi-view setting [58, 47, 44] has become popular. Notably, in multi-view depth estimation, Yifan et al. [57] effectively combined geometric epipolar embeddings with image features for stereo depth estimation, outperforming traditional methods that depend on explicit geometric constraints. State-of-the-art work by [23] integrated multi-view geometric embeddings with image features for video depth estimation. These methods show that the implicit multi-view geometry learned by the Perceiver IO architecture, which is a more efficient general architecture compared to the vision transformer [14], can improve upon approaches that rely on traditional explicit geometric constraints, such as cost volumes, bundle adjustment, and projective geometry. However, the implicit multi-view geometry promoted by the Perceiver IO architecture lacks equivariance. This limitation becomes apparent when transforming the coordinate frame representing input geometry, such as camera poses, ray directions, or 3D coordinates. These transformations change the input in such a way that non-equivariant architectures are unable to achieve the same results, as shown in Figure 1.

Although [23] have tried to approximate equivariance through extensive data augmentation, achieving true equivariance at an architectural level remains an ongoing challenge. In this paper, we propose to embed equivariance with respect to $SE(3)$ transformation of the global coordinate frame, i.e., gauge equivariance, to the Perceiver IO model. We substitute traditional Fourier positional encodings for the ray embedding with Spherical Harmonics, which are more suitable to represent 3D rotations. We custom-develop a $SE(3)$ equivariant attention module to seamlessly interact with different types of equivariant features. This is achieved using a combination of invariant attention weights and equivariant fundamental layers. During the decoding stage, this equivariant latent space is disentangled into the equivariant frame and invariant global features. Our approach not only simplifies the integration of existing modules without requiring a specialized design, but also allows the network to focus on effective scene analysis via an invariant latent space, reducing the effects of global transformations. The equivariant frame is used to "standardize" the query ray, serving as an invariant input for the decoder. This method ensures that both sets of inputs for the decoder are invariant, leading to an invariant output regardless of the decoder used. Consequently, we can employ the conventional Perceiver IO decoder in our equivariant framework. In summary, our key contributions are as follows:

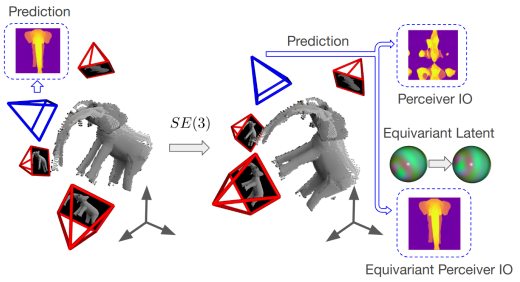


Figure 1: Given a sparse set of posed images (red), the task is to estimate depth for a novel viewpoint (blue). The Perceiver IO struggles to accurately predict depth when the reference frame (gray) changes, equivalent to an inverse transformation applied to the object and cameras. In contrast, our model delivers the consistent result due to its equivariant design.

- We integrate $SE(3)$ equivariance into a multi-view depth estimation model by design, using spherical harmonics as positional encodings for ray embeddings, as well as a specialized equivariant encoder.
- By leveraging the equivariant learned latent space, we introduce a novel scene representation scheme for multi-view settings, featuring a disentangled equivariance frame and an invariant scene representation.
- We assess our model's ability to learn 3D structures through wide-baseline stereo depth estimation. Our model delivers state-of-the-art results in this task, significantly surpassing the non-equivariant baseline.

2 Related Work

Equivariant Networks Equivariant Networks are garnering interest in vision for their efficiency and powerful inductive bias. These networks can be categorized by the data structures over which they operate, spanning 2D images [15, 39], graphs [45, 38], 3D point clouds [60, 5], manifolds [9, 37], and spherical images [17, 7]. From an architectural perspective, methods can also be classified by the

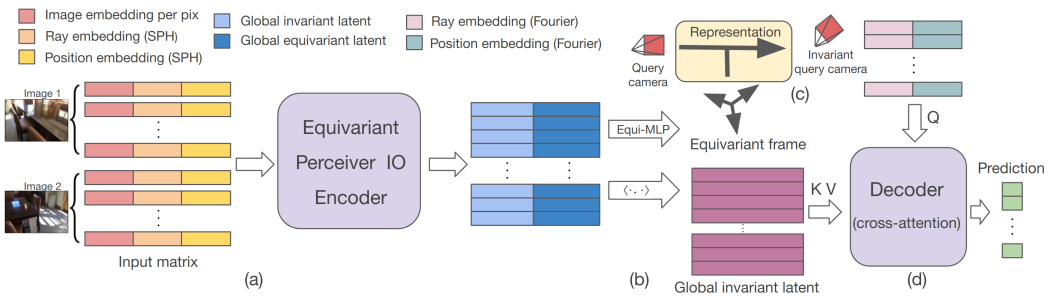


Figure 2: Our proposed Equivariant Perceiver IO (EPIO) architecture. (a) We take as input the concatenation of per-pixel image, ray, and camera embeddings, the latter two calculated using spherical harmonics. (b) The output of our equivariant encoder is a global latent code, including both global invariant and equivariant components. From those, we extract an equivariant reference frame through an equivariant MLP, while simultaneously obtaining invariant latents through inner product. (c) When a query camera is positioned in this equivariant reference frame, its pose becomes invariant, which enables the use of conventional Fourier basis to encode it. (d) Given an invariant latent and invariant pose, we use a conventional Perceiver IO decoder to generate predictions for each query ray.

tools they rely on, such as group convolution [8, 16, 36, 20], steerable convolution on homogeneous spaces [53, 51, 52, 11, 54], and recently transformers [41, 40, 33, 24]. In the context of this paper, we highlight significant $SE(3)$ equivariant transformer works. Fuchs et al. [22] first introduced an SE(3) equivariant transformer for point clouds, using steerable kernels for transformers and focusing on local features in point clouds. Liao and Smidt [33], Liao et al. [34] adopted a message-passing architecture for 3D equivariant transformers in point clouds. Xu et al. [55] applied similar techniques for ray space. Our approach differs by using direct input-level positional encodings, rather than modifying the kernel with relative poses. We learn a global, non-hierarchical representation. Safin et al. [43] inserts pairwise relative poses in self-attention with a conventional attention module, requiring quadratic computation and lacking compact scene representation, unlike our method. Closely related to our work, Assaad et al. [1] proposed vector neuron transformers embedded in the Perceiver IO encoder for point clouds. However, they replace the original latent array with a learnable transformation, did not use spherical harmonics for equivariant positional encoding, or design an equivariant decoder within the Perceiver IO framework. We treat the original latent array as invariant, and learn a disentangled representation for the decoder with versatile queries. Esteves et al. [18] uses spherical harmonics for positional encoding, primarily to enhance spherical function learning, not for equivariance.

Implicit Multi-View Geometry Even in the age of deep learning, traditional multi-view stereo methods like COLMAP [21] are still widely used for structure-from-motion. These methods are accurate but slow due to complex post-processing steps. To speed things up while maintaining accuracy, learning-based methods adapt traditional cost volume-based techniques for depth estimation [29, 3, 26, 27]. Recently, transformers [50] have become prevalent approaches, replacing CNNs in terms of popularity and performance. The Stereo Transformer [32] replaces cost volumes with an attention-based matching procedure inspired by sequence modeling. IIB [57] leverages Perceiver IO [28] for generalized stereo estimation by incorporating the epipolar geometric bias into the model. Liu et al. [35], Chen et al. [4] inject 3D geometry into the transformer akin to IIB for object detection, while Chen et al. [4] learns equivariance in a data-driven way. A closely related study to ours is DeFiNe [23], in which camera information is incorporated into Perceiver IO and used to decode predictions from arbitrary viewpoints. However, their approach relies on data augmentation to approximate equivariance in the Perceiver IO, whereas our design inherently ensures equivariance at an architectural level.

3 Method

In this section we start with some preliminaries about Perceiver IO and our baseline, Depth Field Networks (DeFiNe) [23], a state-of-the-art method integrating camera geometries into Perceiver IO for multi-view depth estimation. We then outline the concept of equivariance in multi-view scenarios in Section 3.2. Given these preliminaries, in Section 3.3 we delve into the details of our proposed equivariant positional encoding for rays, in Section 3.4 we elaborate on the attention mechanisms used in our model, and in Section 3.5 we describe our choice of encoder parameterization. Finally,

in Section 3.6 we describe our decoder procedure, focusing on the task of depth estimation. The pipeline of our proposed $SE(3)$ equivariant model in multi-view context is shown in Figure 2.

3.1 Preliminaries: Input-level Inductive Biases to Perceiver IO

The Perceiver IO [28] is a generalist transformer architecture that encodes input data $\mathcal{I} \in \mathbb{R}^{N_i \times C_i}$ into a latent space $\mathcal{R} \in \mathbb{R}^{N_R \times C_R}$ by cross-attending $\mathcal{I}$ with $\mathcal{R}$. Further refinement of this latent space $\mathcal{R}$ is achieved using self-attention layers, followed by cross-attention to decode predictions $\mathcal{O} \in \mathbb{R}^{N_o \times C_o}$ using queries $\mathcal{Q} \in \mathbb{R}^{N_o \times C_q}$. Many works exploit its generic nature by introducing inductive biases at an input level, namely, providing prior knowledge about the data for implicit reasoning. Specifically, DeFiNe [23] uses camera geometries to construct 3D positional encodings for the multi-view problem. Given N images $\{I_i\}_{i=1}^N$ from a set of cameras with poses $\{T_i\}_{i=1}^N$ and intrinsics $\{K_i\}_{i=1}^N$, DeFiNe calculates 3D rays $\{r_{uv}^i\}_{uv=(1,1)}^{(H,W)}$ from each camera center t_i to each pixel (u, v) on image I_i, and obtains positional encodings $PE(r_{uv}^i, t_i)$ with a mapping $PE(\cdot)$. These positional encodings are combined with image embeddings $\mathcal{F} = \{f_{uv}^i\}$ from a visual feature extractor to be encoded by $\mathcal{R}$ such that:

$$\mathcal{R}_1 = \text{cross-attn}(\mathcal{R}_0, \{f_{uv}^i \oplus PE(r_{uv}^i, t_i)\})$$
$$\mathcal{R}_k = \text{self-attn}(\mathcal{R}_{k-1}), \quad k = 2, \ldots, K$$

To obtain predictions for a set of M novel viewpoints, we can similarly calculate 3D query rays from poses $\{T'_j\}_{j=1}^M$ and intrinsics $\{K_j\}_{j=1}^M$ and map them to query positional encodings $\mathcal{Q} = \{PE(r_{uv}^j, t_j)\}$, which will be used to decode the latent space $\mathcal{R}$ via cross attention: $\mathcal{O} = \text{cross-attn}(\mathcal{Q}, \mathcal{R}_K)$. In this way, prior knowledge, i.e., 3D camera geometries, is directly fed into the model as additional input features for the implicit learning of multi-view geometry.

3.2 Equivariance Definition in Multiview Context

After introducing the input-level inductive bias framework, it is worth noting that the poses of the encoding cameras, as well as the query viewpoints, are defined in a global reference frame T_{G}. However, this choice of global reference frame is subject to change, and the property of equivariance ensures that predictions remain identical under these changes. Assuming the global reference frame undergoes a transform $T^{-1} \in SE(3)$ to $T'_{\text{G}} = T^{-1}T_{\text{G}}$, the ray representations become $(Rr_{uv}^{i(j)}, Rt_{i(j)} + t)$ when representing $T = (R, t)$. The equivariant model Φ should satisfy

$$\Phi(\{f_{uv}^i \oplus PE(Rr_{uv}^i, Rt_i + t)\}, \{PE(Rr_{uv}^j, Rt_j + t)\})$$
$$= \Phi(\{f_{uv}^i \oplus PE(r_{uv}^i, t_i)\}, \{PE(r_{uv}^j, t_j)\}).$$

For further details on the definition of the equivariance, please see Appendix A.2.

3.3 Equivariant Positional Encoding

To ensure the positional encoding process is equivariant w.r.t a transformation group G, we would like to enforce that $\Phi(\cdot, PE(\rho_g^{\mathcal{X}} x)) = \Phi(\cdot, \rho_g^{\mathcal{Y}} PE(x))$ for any $g \in G$, where $\rho^{\mathcal{X}}$ is the group representation on coordinate space, and $\rho^{\mathcal{Y}}$ is the group representation on the positional encoding space. The traditional Fourier basis is translationally equivariant, as detailed in Appendix B. Kitaev et al. [30] used this to attain translational equivariance, employing a conjugate product for invariant attention. However, this approach lacks rotational equivariance.

This raises a key question: Are there any basis functions equivariant to both 3D translations and rotations? Unfortunately, none exist. However, a common method in equivariant works for translational equivariance is to subtract the center point, a technique we apply in our context as illustrated in Figure 3. This enables translational invariance, leaving the model to focus solely on achieving rotational equivariance. To address the 3D rotational equivariance, we turn to spherical harmonics (SPH), known for their inherent rotation-equivariant properties. They offer a way to accommodate 3D rotational changes, thereby achieving $SE(3)$ equivariance.

3.3.1 Spherical Harmonics

We provide a detailed introduction to Spherical Harmonics in Appendix A.4, where we also discuss how their application in previous equivariant transformers differs from their use in our model. Below,

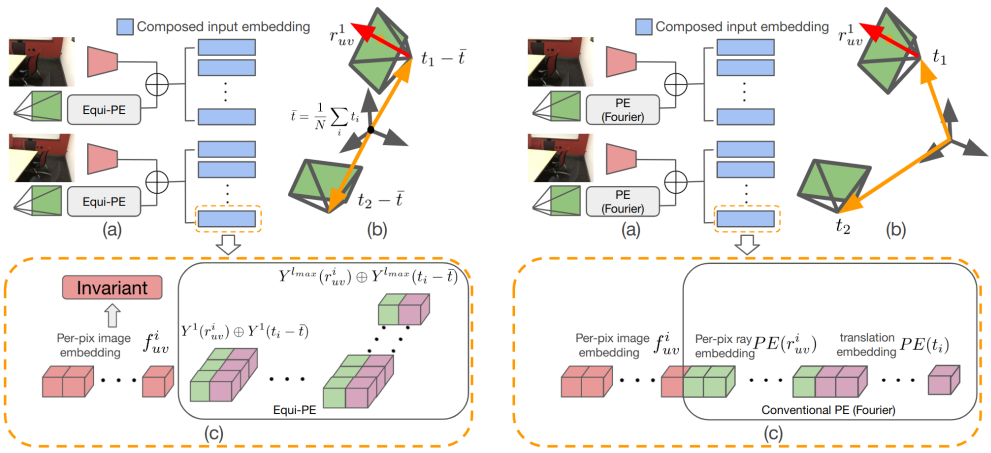


Figure 3: Comparison between an equivariant input embedding in our model (left) and the conventional input embedding in DeFiNe (right). (a) Pipeline used to generate input embeddings for the encoder, resulting in cross-attention keys and values. (b) To generate geometric information, we calculate embeddings for each ray r^i_{uv} and camera relative position $t_i - \bar{t}$; (c) The final composed embedding format includes both image embeddings, which are invariant, and geometric embeddings, which are equivariant. In contrast, the conventional approach by Perceiver IO, as highlighted in parts (a) and (c), integrates Fourier positional encodings with image embeddings to form the input embeddings. Furthermore, as indicated in (b), Perceiver IO utilizes each ray r^i_{uv} and the absolute translation t_i for positional encoding purposes.

we present a brief overview of Spherical Harmonics for clarity. Similar to the varying frequencies of sines and cosines in Fourier series, spherical harmonics are characterized by different degrees (orders), denoted as $l \in \mathbb{N}$. An order-l spherical harmonics, denoted as $Y^l : \mathbb{R}^3 \to \mathbb{R}^{2l+1}$, follows the transformation rule: $Y^l(Rr) = D^l(R)Y^l(r), Y^l(r) = \|r\|^l Y^l(\hat{r})$, where $R \in SO(3)$, $\hat{r}$ is the unit vector, $D^l : SO(3) \to \mathbb{R}^{(2l+1)\times(2l+1)}$ is called the Wigner-D matrix, serving as the irreducible representation of $SO(3)$ corresponding to the order l. The Wigner-D matrix is an orthogonal matrix, that is $D^l(R)(D^l(R))^T = I$. These important properties allow us to achieve equivariance in the Perceiver IO transformer architecture.

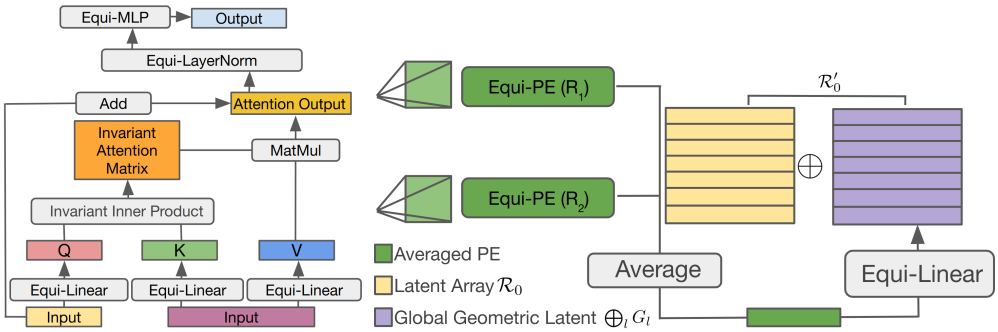
3.3.2 Equivariant Hidden features

In our model, we embed both camera centers and viewing rays using spherical harmonics. The embedding is given by: $PE(r^i_{uv}, t_i) = \bigoplus_{l \in L}(Y^l(r^i_{uv}) \oplus Y^l(t_i - \bar{t}))$, where each part corresponds to the same order of spherical harmonics (Figure 3). Here, $L = \{1, 2, \ldots, l_{max}\}$ and $\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i$, highlighting the extraction of the cameras' central position for translational invariance. Due to the properties of spherical harmonics, the positional encoding of transformed input, $PE(Rr^i_{uv}, Rt_i + t)$, is equal to $\bigoplus_{l \in L}(D^l(R)(Y^l(r^i_{uv}) \oplus Y^l(t_i - \bar{t}))) = R \cdot PE(r^i_{uv}, t_i)$, for any rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$. In other words, it guarantees that these embeddings are both *rotationally equivariant* and *translationally invariant*. The transformation of these embeddings operates by multiplying each block with its respective Wigner-D matrix.

The image remains unchanged when the reference frame is transformed, as its contents are unaffected. Mathematically, this property is akin to multiplying by a 0-order Wigner Matrix, equivalent to an identity. Thus, combining image features with positional encodings (Figure 3) transform as:

$$\begin{aligned}(f^i_{uv}, PE(Rr^i_{uv}, Rt_i + t)) &= (D^0(R)f^i_{uv}, R \cdot PE(r^i_{uv}, t_i)) \\ &= R \cdot (f^i_{uv}, PE(r^i_{uv}, t_i))\end{aligned}$$

In our model, the equivariant hidden features mirror the structure of our embeddings, composed as $\bigoplus_{l \in L} H_l$ with subscripts indicating the feature type and $L = \{0, 1, \cdots, l_{max}\}$. The size for each feature type H_l follows $(2l + 1, C_l)$, where $2l + 1$ is the intrinsic dimension and C_l is the number of channels. For more a intuitive understanding, we visualize these embeddings in Appendix C (Figure 10). Similar to the input embeddings, any rotation R in $SO(3)$ rotates the hidden features as



(a) (b)

Figure 4: Left: Our equivariant module is distinct from traditional implementations [50] in its fundamental layers and the key-query product, that are crafted to be respectively equivariant and invariant. Right: Equivariant latent array used as additional input to the encoder. We apply equivariant positional encoding to each camera rotation, which is then averaged. We leverage an equivariant linear layer to get a global geometric latent $\oplus_l G_l$, which is concatenated with the conventional latent array $\mathcal{R}_0$ to compose our proposed equivariant latent array $\mathcal{R}_0'$.

$$R \cdot \bigoplus_{l \in L} H_l = \bigoplus_{l \in L} D^l(R) H_l$$

where D^l are the Wigner-D matrices. We disregard any translation action since the input and queries become translation-invariant after center subtraction.

3.4 Basic Attention Modules

This section highlights the equivariant attention module, fundamental to ensure encoder equivariance as depicted in Figure 4a. It consists of basic equivariant layers and an invariant multi-head inner product. Our architecture, unlike typical equivariant transformers [22, 33], does not enforce geometric constraints in the equivariant kernel. Instead, it incorporates all geometric features at an input-level. Our module, utilizing the Perceiver IO structure, learns global latent representations, in contrast to other methods that emphasize the hierarchical learning of local features.

Equivariant Foundamental Layers For the fundamental layers, we use the equivariant linear layer and layer normalization commonly used in previous works [48, 22, 33, 34], and provide additional details in Appendix A.3. For equivariant nonlinear layers, there have been multiple proposed methods for features with the same format as ours: Norm-based Nonlinearity, Gate Nonlinearity, and Fourier-based Nonlinearity. Here, we take inspiration from the nonlinearity of Vector Neuron [13] and adapt a similar vector operation to higher-order features. Please see Appendix E for details of equivariant nonliearity. To better understand the differences between the basic layers in equivariant attention module and those in conventional one, we have visualize them in Appendix A.3 and Appendix E.

Multi-Head Attention Inner Product As done in previous equivariant transformer works [22, 33], we can obtain the invariant attention matrix through inner product of the same types of features. These transformers that emphasize the hierarchical learning of local features suggest using tensor products of edge feature and node feature to mix different feature types, which is computationally demanding. We discard the tensor product and only calculate attention weights using various feature types and then multiply these weights with multi-type features to efficiently integrate different types of feature. Please see Appendix F for more details. Alternatively, we can mix feature types by treating them as Fourier coefficients for spheres, apply transformers on the sphere, and use Fourier Transform to obtain new coefficients. Please refer to Appendix H for details.

3.5 Equivariant Encoder

3.5.1 Equivariant Cross-attention

As shown in the left part in Figure 3 and Section 3.3.2, the cross-attention input is in the format $\bigoplus_{l\in L} H_l$ with $C_l = 2$ for $l \geq 1$ and C_0 being the channel number of f^i_{uv}. To facilitate a clearer understanding, a comparison of this input embedding with the one used in DeFiNE is also depicted in Figure 3. The latent array $\mathcal{R}_0 = ((R_0)_1, (R_0)_2, \cdots, (R_0)_{N_R}) \in \mathbb{R}^{N_R \times C_R}$ can be treated as a scalar (0-order) feature, remaining constant during transformations in the reference frame. To make the latent array also learn the geometric information, we apply a technique similar to [1], learning equivariant features from the input's averaged geometric information. Specifically, we apply the positional encoding (PE) for each camera rotation, with each order being the concatenation of embeddings of the rotation matrix's three columns. The PE is then averaged over cameras. For the specific formulation please see Appendix I.

We obtain a global geometric latent $\mathcal{G}$ using an equivariant linear layer, where the size of the weight matrix W_l for each type l is $(3, N_R C^l)$, with C^l being the channel count for type-l feature in each latent. We then append this equivariant feature to the latent $\mathcal{R}_0$, forming a new latent array $\mathcal{R}'_0 = ((R'_0)_1, (R'_0)_2, \cdots, (R'_0)_{N_R})$, where $(R'_0)_i = (R_0)_i \oplus \bigoplus_{l\in L}(G_l)_i$ with $L = \{1, 2, \cdots l_{max}\}$ and $(G_l)_i \in \mathbb{R}^{(2l+1)\times C^l}$. Figure 4b illustrates the construction of this equivariant latent array. With both an equivariant input embedding and latent array, we apply equivariant cross-attention to get the equivariant latent output.

3.5.2 Equivariant Self-Attention

We apply a self-attention equivariant attention mechanism to the equivariant output of cross-attention, producing a conditioned equivariant latent code. For visualization purposes (Figure 5), we can treat the equivariant latent code as the Fourier coefficients of spherical functions. Note that we do not map the features to a 2D color image. Since we have features with type-0, type-1, and type-2, etc, but we randomly select each channel from different types of feature and apply the inverse Fourier transform to get a spherical function and visualize it on a 3D sphere. For a proof of this result (i.e., the visualized sphere is also rotated when the latent code is rotated), please see Appendix D.

3.6 Decoder

In Figure 2 we show that, before inputting the equivariant latent space and geometric query to the decoder, they are converted into an invariant form by establishing an equivariant frame. Specifically, the equivariant latent space $\mathcal{R}_K$ is represented as $\bigoplus_{l\in L}(\mathcal{R}_K)_l$. From its type-1 feature $(\mathcal{R}_K)_1$, we employ an equivariant MLP and the Gram-Schmidt orthogonalization [59] to derive an equivariant frame, represented by a rotation matrix R. As depicted in Figure 5, the equivariant frame's rotation aligns with that of both the equivariant latent and the $3D$ scene. Applying the inverse of R to $\mathcal{R}_K$ results in a rotation-invariant latent code $\bigoplus_{l\in L} D^l(R)^T(\mathcal{R}_K)_l$, obtaining an invariant representation. See Appendix J for a proof.

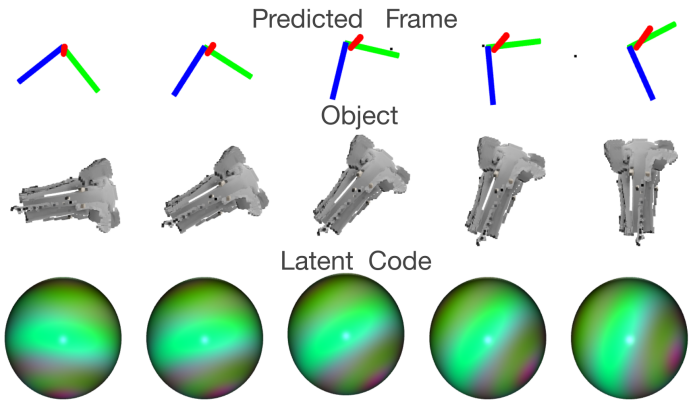


Figure 5: Equivariant latent code and predicted frame. For simplicity, we use object rotation to denote the inverse rotation of the reference frame. When the object is rotated, our latent code and predicted canonical frame are also rotated.

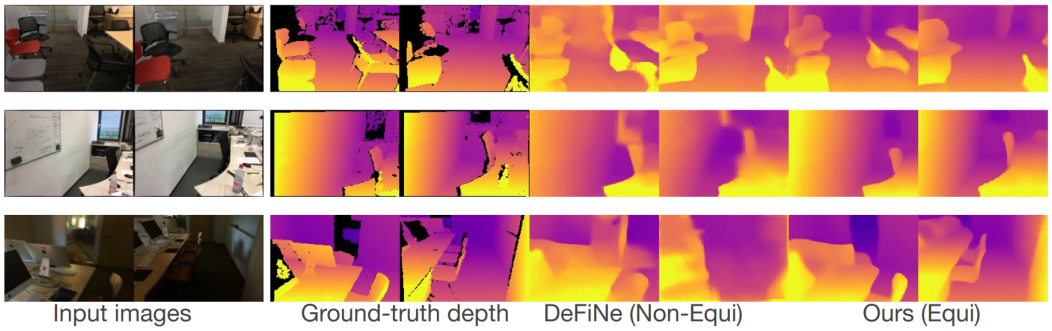


Figure 6: Stereo depth estimation results on ScanNet, using our proposed EPIO architecture.

For the embedding of camera center and viewing rays, we apply R^T to r^j_{uv} and $t_j - \bar{t}$ to obtain invariant coordinates (see Appendix J for a proof), denoted as $R^T r^j_{uv}$ and $R^T(t_j - \bar{t})$. We then use traditional sine and cosine positional encoding for these invariant coordinates, which allows us to leverage higher frequency information beyond the dimensional constraints of SPH. Since both the latent hidden state and the query are invariant to the transformation, this enables us to apply conventional cross-attention mechanisms to obtain invariant outputs and predictions, capturing higher frequency details and improving expressiveness.

4 Experimental Results

4.1 Datasets and Implementation

We use **ScanNet** [12] and **DeMoN** [49] to validate our model on the task of stereo depth estimation. For ScanNet, we use the same setting as [31], which downsamples scenes by a factor of 20 and splits them to obtain 94212 training and 7517 test pairs. The DeMoN dataset includes the SUN3D, RGBD-SLAM and Scenes11 datasets, where SUN3D and RGBD-SLAM are real world datasets and Scenes11 is a synthetic dataset. There are a total of 166285 training image pairs from 50420 scenes, and we use the same test split as [31] (80 pairs in SUN3D, 80 pairs in RGBD and 168 pairs in Scenes11). We include details on the network architecture and implementation in Appendix L.

4.2 Stereo Depth Estimation

We compare our equivariant model with other state-of-the-art methods on stereo depth estimation, and report quantitative results in Table 1. As we can see, it significantly outperforms competing methods on all real-world datasets and shows comparable results to the state-of-the-art on Scenes11, a synthetic dataset. This superior performance on real-world datasets is evidence of the benefits of using equivariance in multi-view scene representation. Synthetic datasets, unaffected by real-world lighting conditions, camera miscalibration and view-dependent artifacts, might benefit approaches such as DPSNet [27] and NAS [31] that use cost volume to achieve view consistency. It's also noteworthy that NAS [31] uses additional ground truth surface normals as supervision.

We denote our model with "Equi" and our baseline, DeFiNe, with "Nonequi" to highlight that both use the same architecture, Perceiver IO, with the key difference being the presence of equivariance in our model.

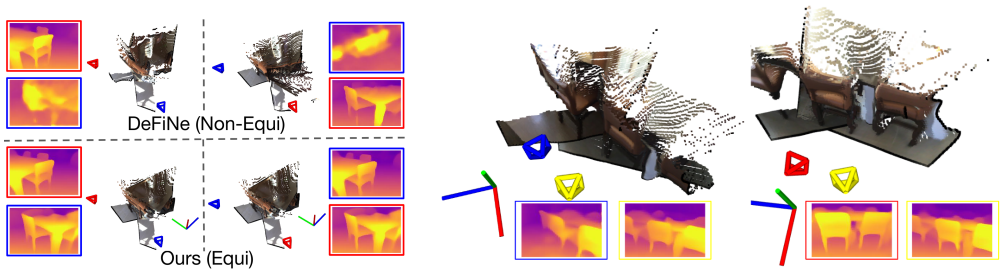
Method	Abs.Rel. ↓	RMSE ↓	$\delta < 1.25$ ↑
DeFiNe (w/o VCA)	0.117	0.291	0.870
Ours (w/o VCA)	0.104	0.247	0.893
DeFiNe (w/o jitter)	0.099	0.261	0.891
DeFiNe	0.093	0.246	0.911
Ours (w/o jitter)	**0.086**	**0.229**	**0.923**

Table 2: Comparison of our EPIO model and DeFiNe on ScanNet regarding the use of data augmentation. *VCA* stands for *virtual camera augmentation*, and *jitter* stands for *canonical camera jittering*.

Additionally, to assess the advantages of incorporating equivariance into our model, we conducted a comparative analysis of our model against our nonequivariant DeFiNe baseline [23], both with and without

Dataset	Method	Abs.Rel. ↓	RMSE ↓	$\delta < 1.25$ ↑	Dataset	Method	Abs.Rel. ↓	RMSE ↓	$\delta < 1.25$ ↑
ScanNet	DPSNet	0.126	0.315	-	RGBD-SLAM	DeMoN	0.157	1.780	0.801
	NAS	0.107	0.281	-		DeepMVS	0.294	0.868	0.549
	IIB	0.116	0.281	0.908		DPSNet	0.151	0.695	0.804
	DeFiNe	0.093	0.246	0.911		NAS	0.131	0.619	0.857
	Ours	**0.086**	**0.229**	**0.923**		IIB	0.095	0.550	0.907
						Ours	**0.080**	**0.433**	**0.912**
SUN3D	DeMoN	0.214	2.421	0.733	Scenes11	DeMoN	0.556	2.603	0.496
	DeepMVS	0.282	0.944	0.562		DeepMVS	0.210	0.891	0.688
	DPSNet	0.147	0.449	0.781		DPSNet	0.050	0.466	0.961
	NAS	0.127	0.378	0.829		NAS	**0.038**	**0.371**	**0.975**
	IIB	0.099	0.293	0.902		IIB	0.055	0.523	0.963
	Ours	**0.090**	**0.260**	**0.912**		**Ours**	0.069	0.617	0.965

Table 1: Stereo depth estimation results compared with the state-of-the-art: DPSNet [27], NAS [31], IIB [57], DeFine [23], DeMoN [49], DeepMVS [26].



(a) (b)

Figure 7: The figure (a) shows the equivariance of changing reference frame (Red: reference frame): For the same input and varying camera frames as reference, the Perceiver IO's predictions change, but our model's predictions stay consistent and the predicted frame is equivariant to the reference frame transformation. The figure (b) shows the approximate equivariance for different camera sets.

data augmentation. We explore two kinds of data augmentation: virtual camera augmentation (VCA), in which novel viewpoints are generated via pointcloud reprojection; and canonical camera jittering (CCJ), in which the reference frame is perturbed with random rotation and translation, reported in Table 2.

To further showcase our equivariant properties, we visualize the predicted canonical frame and reconstructed $3D$ point clouds from depth maps. In Figure 7a, we see that, for the same scene, when we switch the reference frame (in red) between cameras, the output point clouds change when using the standard Perceiver IO architecture, while ours remain constant, since the predicted depth is equivariant to transformations. Furthermore, even when we use different image pairs within the same scene, which theoretically cannot be guaranteed equivariant due to changes in image content, our model still predicts near-consistent canonical frames and point clouds, as illustrated in Figure 7b.In the meanwile, we compare our model with current state-of-art depth estimation model Depth anything [56], and we provide the results in Appendix O.1.

4.3 Ablation Study

We performed an ablation study on the geometric positional encodings, spherical harmonics encoding, equivariant attention, and the decoder architecture, and report the quantitative results in Table 3. As expected, when positional encoding is not used, results are significantly degraded due to missing geometric information. Our results demonstrate that the model leverages geometric information to learn implicit multi-view geometry. Although using Fourier positional encodings with our method breaks the equivariant properties, we conducted ablations by replacing spherical harmonics with Fourier encodings to assess the specific contribution of spherical harmonics.

As shown in the table, Fourier encodings, which are not equivariant, are incompatible with an equivariant architecture, resulting in significantly worse performance. Additionally, we replaced the equivariant attention module with a conventional one in another ablation study. Removing the equivariant attention layers also disrupts the equivariance of our architecture, leading to a substantial drop in performance since the model loses its theoretical equivariance. We also explore the impact of the maximum order of spherical harmonics in the positional encodings, indicated by l_{max}. For network architecture, we evaluated the impact of not learning the canonical frame, that is, we use the equivariant attention module in the decoder followed by transferring the equivariant output to invariant output via inner product, see Appendix K for details. We noticed that higher order of spherical harmonics improve depth estimation, since high frequency promotes fine-grained learning and differentiate positions in a higher-dimensional space. Unlike Fourier basis, the dimension of the spherical harmonics grows two times linearly with increasing order, which is a limitation of our method, and therefore we keep the highest SPH order as 8 in our final model. This is also a reason why learning a equivariant canonical frame for invariant decoding with Fourier basis and a conventional decoder is a better approach than directly using an equivariant decoder. Another factor is that learning a canonical frame enforces all inputs to the decoder to be invariant, which should facilitate 3D reasoning. Moreover, we performed additional small-scale experiments to study the impact of the number of available views, see Appendix O.2.

Variation	Abs.Rel.↓	RMSE↓	$\delta < 1.25 \uparrow$
w/o camera information	0.229	0.473	0.661
w/ Fourier	0.131	0.318	0.843
w/o equi-attention	0.127	0.314	0.851
Type $l_{max} = 1$	0.134	0.310	0.869
Type $l_{max} = 2$	0.125	0.302	0.875
Type $l_{max} = 4$	0.116	0.283	0.898
EquiDecoder	0.128	0.317	0.857
Full Model	**0.086**	**0.229**	**0.923**

Table 3: Ablation study on the choice of positional encoding frequency and decoder architecture.

5 Conclusion and Discussion

We introduce an $SE(3)$ equivariant model designed to learn the equivariant $3D$ scene prior across multiple views, utilizing spherical harmonics for positional encoding and specialized equivariant attention mechanisms within the Perceiver IO architecture. Additionally, our design exploits its equivariant latent space to disentangle equivariant frames and invariant scene details, enabling the seamless integration of various existing decoders in conjunction with our specialized encoder. our model's capability in 3D structure comprehension is showcased through its superior performance in stereo depth estimation, significantly exceeding that of non-equivariant models. Our architecture can be modified to accommodate a wider range of vision tasks, which we leave to future work (for a more detailed discussion please see Appendix M).

Limitation As discussed in Section 4.3, unlike the Fourier basis, the dimension of spherical harmonics increases linearly at twice the rate with each order. This limits the number of spherical harmonics and the maximum frequency utilized, resulting in an inability to preserve detailed features in cameras and images. Additionally, the presence of different types of features, each with its own linear and nonlinear layers, slightly slows down the forward process compared to traditional methods. Moreover, we observe instability in training the equivariant network, which may be due to the magnitude explosion of high-order spherical harmonics.

Acknowledgment

This research was supported by Toyota Research Institute, whose funding and resources were invaluable in advancing this work.

References

[1] Serge Assaad, Carlton Downey, Rami Al-Rfou, Nigamaa Nayakanti, and Ben Sapp. Vn-transformer: Rotation-equivariant attention for vector neurons. *arXiv preprint arXiv:2206.04176*, 2022.

[2] Johann Brehmer, Joey Bose, Pim De Haan, and Taco Cohen. Edgi: Equivariant diffusion for planning with embodied agents. *arXiv preprint arXiv:2303.12410*, 2023.

[3] Joao Carreira, Skanda Koppula, Daniel Zoran, Adria Recasens, Catalin Ionescu, Olivier Henaff, Evan Shelhamer, Relja Arandjelovic, Matt Botvinick, Oriol Vinyals, et al. Hip: Hierarchical perceiver. *arXiv preprint arXiv:2202.10890*, 2022.

[4] Dian Chen, Jie Li, Vitor Guizilini, Rares Andrei Ambrus, and Adrien Gaidon. Viewpoint equivariance for multi-view 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9213–9222, 2023.

[5] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3d point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14514–14523, 2021.

[6] Yunlu Chen, Basura Fernando, Hakan Bilen, Matthias Nießner, and Efstratios Gavves. 3d equivariant graph implicit functions. In *European Conference on Computer Vision*, pages 485–502. Springer, 2022.

[7] Oliver J Cobb, Christopher GR Wallis, Augustine N Mavor-Parker, Augustin Marignier, Matthew A Price, Mayeul d'Avezac, and Jason D McEwen. Efficient generalized spherical cnns. *arXiv preprint arXiv:2010.11661*, 2020.

[8] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[9] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pages 1321–1330. PMLR, 2019.

[10] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

[11] Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32, 2019.

[12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[13] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. *arXiv preprint arXiv:1709.01889*, 2017.

[16] Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1568–1577, 2019.

[17] Carlos Esteves, Ameesh Makadia, and Kostas Daniilidis. Spin-weighted spherical cnns. *Advances in Neural Information Processing Systems*, 33:8614–8625, 2020.

[18] Carlos Esteves, Tianjian Lu, Mohammed Suhail, Yi-fan Chen, and Ameesh Makadia. Generalized fourier features for coordinate-based learning of functions on manifolds. 2021.

[19] Carlos Esteves, Jean-Jacques Slotine, and Ameesh Makadia. Scaling spherical cnns. *arXiv preprint arXiv:2306.05420*, 2023.

[20] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, pages 3165–3176. PMLR, 2020.

[21] Alex Fisher, Ricardo Cannizzaro, Madeleine Cochrane, Chatura Nagahawatte, and Jennifer L Palmer. Colmap: A memory-efficient occupancy grid mapping framework. *Robotics and Autonomous Systems*, 142:103755, 2021.

[22] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

[23] Vitor Guizilini, Igor Vasiljevic, Jiading Fang, Rare Ambru, Greg Shakhnarovich, Matthew R Walter, and Adrien Gaidon. Depth field networks for generalizable multi-view scene representation. In *European Conference on Computer Vision*, pages 245–262. Springer, 2022.

[24] Lingshen He, Yiming Dong, Yisen Wang, Dacheng Tao, and Zhouchen Lin. Gauge equivariant transformer. *Advances in Neural Information Processing Systems*, 34:27331–27343, 2021.

[25] Haojie Huang, Dian Wang, Xupeng Zhu, Robin Walters, and Robert Platt. Edge grasp network: A graph-based se (3)-invariant approach to grasp detection. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3882–3888. IEEE, 2023.

[26] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018.

[27] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019.

[28] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[29] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE international conference on computer vision*, pages 66–75, 2017.

[30] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[31] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2189–2199, 2020.

[32] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6197–6206, 2021.

[33] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. *arXiv preprint arXiv:2206.11990*, 2022.

[34] Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. *arXiv preprint arXiv:2306.12059*, 2023.

[35] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer, 2022.

[36] Lachlan E MacDonald, Sameera Ramasinghe, and Simon Lucey. Enabling equivariance for arbitrary lie groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8183–8192, 2022.

[37] DE Pim, Maurice Weiler, Taco Sebastiaan Cohen, and Max Welling. Gauge equivariant geometric graph convolutional neural network, August 12 2021. US Patent App. 17/169,338.

[38] Omri Puny, Derek Lim, Bobak Kiani, Haggai Maron, and Yaron Lipman. Equivariant polynomials for graph neural networks. In *International Conference on Machine Learning*, pages 28191–28222. PMLR, 2023.

[39] Md Ashiqur Rahman and Raymond A Yeh. Truly scale-equivariant deep nets with fourier layers. *arXiv preprint arXiv:2311.02922*, 2023.

[40] David Romero, Erik Bekkers, Jakub Tomczak, and Mark Hoogendoorn. Attentive group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 8188–8199. PMLR, 2020.

[41] David W Romero and Jean-Baptiste Cordonnier. Group equivariant stand-alone self-attention for vision. *arXiv preprint arXiv:2010.00977*, 2020.

[42] Hyunwoo Ryu, Hong-in Lee, Jeong-Hoon Lee, and Jongeun Choi. Equivariant descriptor fields: Se (3)-equivariant energy-based models for end-to-end visual robotic manipulation learning. *arXiv preprint arXiv:2206.08321*, 2022.

[43] Aleksandr Safin, Daniel Durckworth, and Mehdi SM Sajjadi. Repast: Relative pose attention scene representation transformer. *arXiv preprint arXiv:2304.00947*, 2023.

[44] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6229–6238, 2022.

[45] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[46] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[47] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pages 156–174. Springer, 2022.

[48] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[49] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[51] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019.

[52] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018.

[53] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5028–5037, 2017.

[54] Yinshuang Xu, Jiahui Lei, Edgar Dobriban, and Kostas Daniilidis. Unified fourier-based kernel and nonlinearity design for equivariant networks on homogeneous spaces. In *International Conference on Machine Learning*, pages 24596–24614. PMLR, 2022.

[55] Yinshuang Xu, Jiahui Lei, and Kostas Daniilidis. $se(3)$ equivariant convolution and transformer in ray space. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[56] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

[57] Wang Yifan, Carl Doersch, Relja Arandjelović, Joao Carreira, and Andrew Zisserman. Input-level inductive biases for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6176–6186, 2022.

[58] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. *arXiv preprint arXiv:2402.14817*, 2024.

[59] 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.

[60] Minghan Zhu, Maani Ghaffari, William A Clark, and Huei Peng. E2pn: Efficient se (3)-equivariant point network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1223–1232, 2023.

Appendix

A Preliminaries

A.1 Perceiver IO

The Perceiver IO [28] efficiently encodes multi-modality inputs by utilizing cross-attention between the inputs themselves and a learnable, fixed-dimension latent code. This latent code is then refined through a series of self-attention layers. In the decoding phase, the model employs cross-attention between a given query and the refined latent code to generate predictions.

In DeFiNe [23], this framework is used to address scenarios involving multiple cameras with predetermined relative poses, denoted as $\{T_i\}_{i=1}^N$, and their respective images $\{I_i\}_{i=1}^N$. Within this context, the system queries arbitrary camera poses, represented as $\{T'_j\}_{j=1}^M$. Importantly, this queried pose T_j can either be one of the already established camera positions (as it is common in stereo depth estimation), or a position outside the range of the input cameras (which is typical in video depth estimation). Based on input data and queried pose, the network generates the corresponding predicted output $\hat{D}_j$ for the query.

Like other vision tasks that utilize the Perceiver framework, DeFiNe uses as input a composite of geometric information and corresponding image features, while the query utilizes only geometric information. Specifically, the input is formulated as $\left\{f^i_{uv} \oplus PE(r^i_{uv}) \oplus PE(t_i)\right\}$, where f^i_{uv} represents image features associated with each pixel (u, v) in camera i. $PE(r^i_{uv})$ denotes the positional encoding with Fourier cosine and sine series of the ray direction r^i_{uv} calculated relative to camera T_i, $PE(t_i)$ refers to the positional encoding of the camera's translation t_i in T_i, using Fourier cosine and sine series. The query in this model is represented as $\left\{PE(r^j_{uv}) \oplus PE(t_j)\right\}$, and the network is designed to output depth estimation $\hat{D^j_{uv}}$ for each query pixel (u, v) of query camera j.

A.2 Equivariance Definition

Concretely, assuming the global reference frame undergoes a transform $T^{-1} \in SE(3)$ to $T'_{\mathrm{G}} = T^{-1}T_{\mathrm{G}}$, the set of input and query poses would become $\{TT_i\}$ and $\{TT'_j\}$ with respect to T'_{G} (note that the corresponding input images $\{I_i\}_{i=1}^N$ remain unchanged). Mathematically, we use $\Lambda_T((\{T_i\}, \{I_i\})) = (\{TT_i\}, \{I_i\})$ and $\Lambda_T(\{T'_j\}) = \{TT_j\}$ to denote the $SE(3)$ actions on the input and query cameras. An equivariant network satifies

$$\Phi(\Lambda_T(\{T_i\}, \{I_i\}), \Lambda_T(\{T'_j\})) \equiv \Phi((\{T_i\}, \{I_i\}), \{T'_j\}).$$

Readers may recognize the above equation as describing the invariance of the network Φ to the transformation of both input and query. In fact, it is also equivalent to the statement that the network Φ is equivariant when viewed that it learns an implicit field. To demonstrate this equivalence, let $F(\cdot) = \Phi((\{T_i\}, \{I_i\}), \cdot)$, and define the $SE(3)$ operator Λ' on F: $\Lambda'_T F(\{T_j\}) = F(\Lambda_T^{-1}(\{T_j\}))$. We then derive

$$\Phi(\Lambda_T(\{T_i\}, \{I_i\})) = \Lambda'_T F = \Lambda'_T \Phi(\{T_i\}, \{I_i\}),$$

i.e., that the network Φ is equivariant. A similar statement can be found in [6].

With input level inductive bias and representing $T = (R, t)$, the equivariant model Φ should satisfy

$$\begin{aligned}&\Phi(\{f^i_{uv} \oplus PE(Rr^i_{uv}, Rt_i + t)\}, \{PE(Rr^j_{uv}, Rt_j + t)\}) \\ &= \Phi(\{f^i_{uv} \oplus PE(r^i_{uv}, t_i)\}, \{PE(r^j_{uv}, t_j)\}).\end{aligned}$$

A.3 Fundamental Layers

A.3.1 Equivariant Linear Layer

With the transformation acting on the features, we can define the equivariant linear layer $\mathcal{L}$:

$$\mathcal{L}(R \cdot H^{(k)}) = R \cdot \mathcal{L}(H^{(k)}) = R \cdot H^{(k+1)},$$

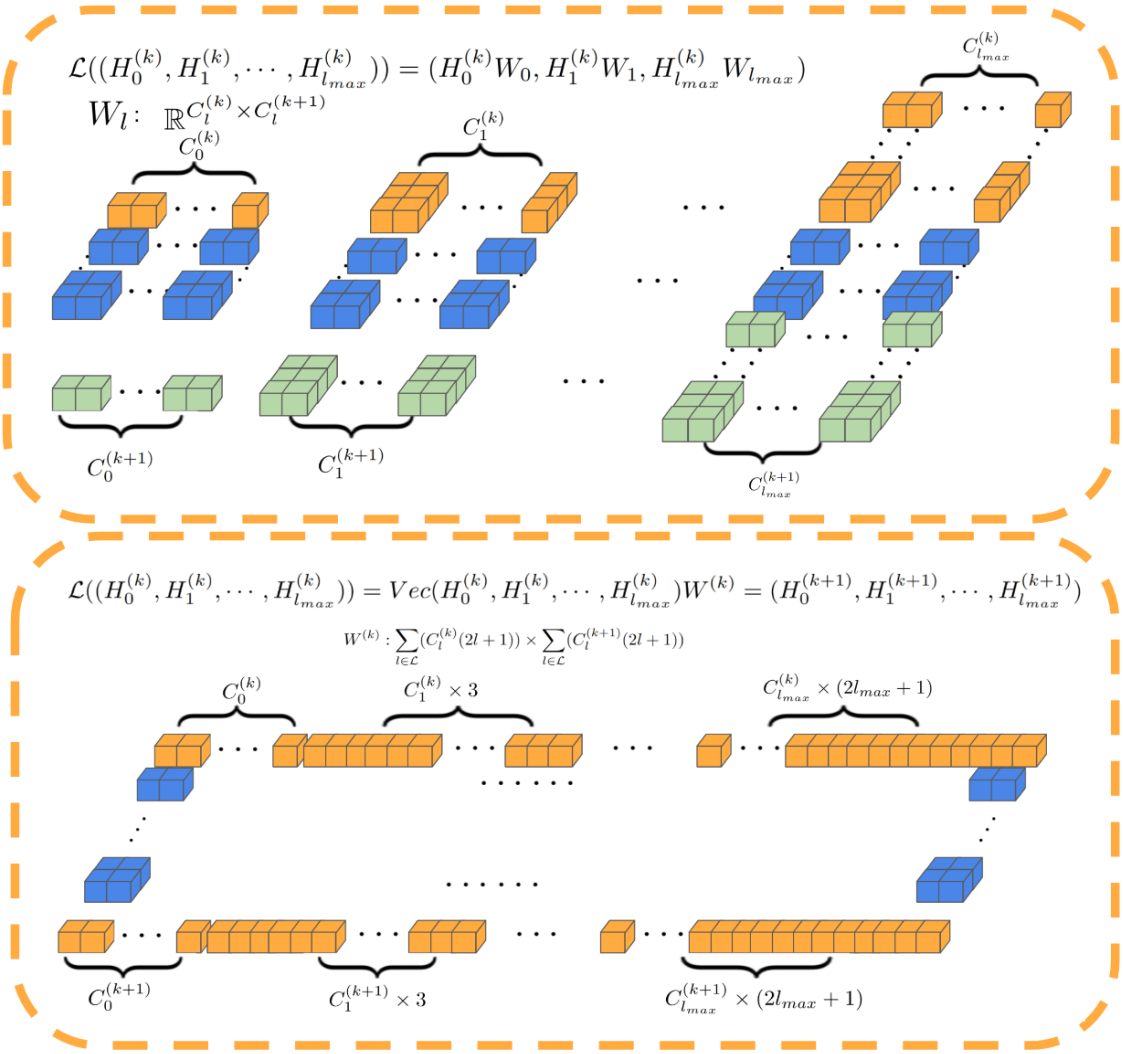


Figure 8: Comparison between Equivariant and Non-Equivariant Linear Layers: The figure above depicts the equivariant linear layer, wherein each type of feature is linearly combined using a specific matrix W_l, treating the vector or tensor as a cohesive geometric entity. Conversely, the figure below illustrates the traditional linear approach, where all channels within an intrinsic feature, as well as different types of features, are linearly intermixed, since it vectorizes and concatenates all features and applies a unified weight matrix

where the superscript denotes the index of layer $H^{(k)} = \bigoplus_{l \in L} H_l^k = (H_0^{(k)}, H_1^{(k)}, \cdots, H_{l_{max}}^{(k)})$ and the same for $H^{(k+1)}$. To achieve equivariance, we use the same linear layer $\mathcal{L}$ as stated in [48, 52, 33]:

$$\begin{aligned}&\mathcal{L}((H_0^{(k)}, H_1^{(k)}, \cdots, H_{l_{max}}^{(k)})) \\ &= (H_0^{(k)}W_0, H_1^{(k)}W_1, H_{l_{max}}^{(k)}W_{l_{max}}).\end{aligned}$$

The weights W_l have the format $(C_l^{(k)}, C_l^{(k+1)})$, where $C_l^{(k)}$ is the number of channels in $H_l^{(k)}$, representing the corresponding input channels, and $C_l^{(k+1)}$ is the number of channels in $H_l^{(k+1)}$, representing the corresponding output channels. The difference of equivariant linear layers and conventional linear layers are depicted in Figure 8.

A.3.2 Equivariant LayerNormalization

We use the commonly used equivariant layer normalization in equivariant works that apply the layer normalization to the norm of the features, and then multiply those with unit tensor features. The normalization layer $\mathcal{LN}$ is defined as

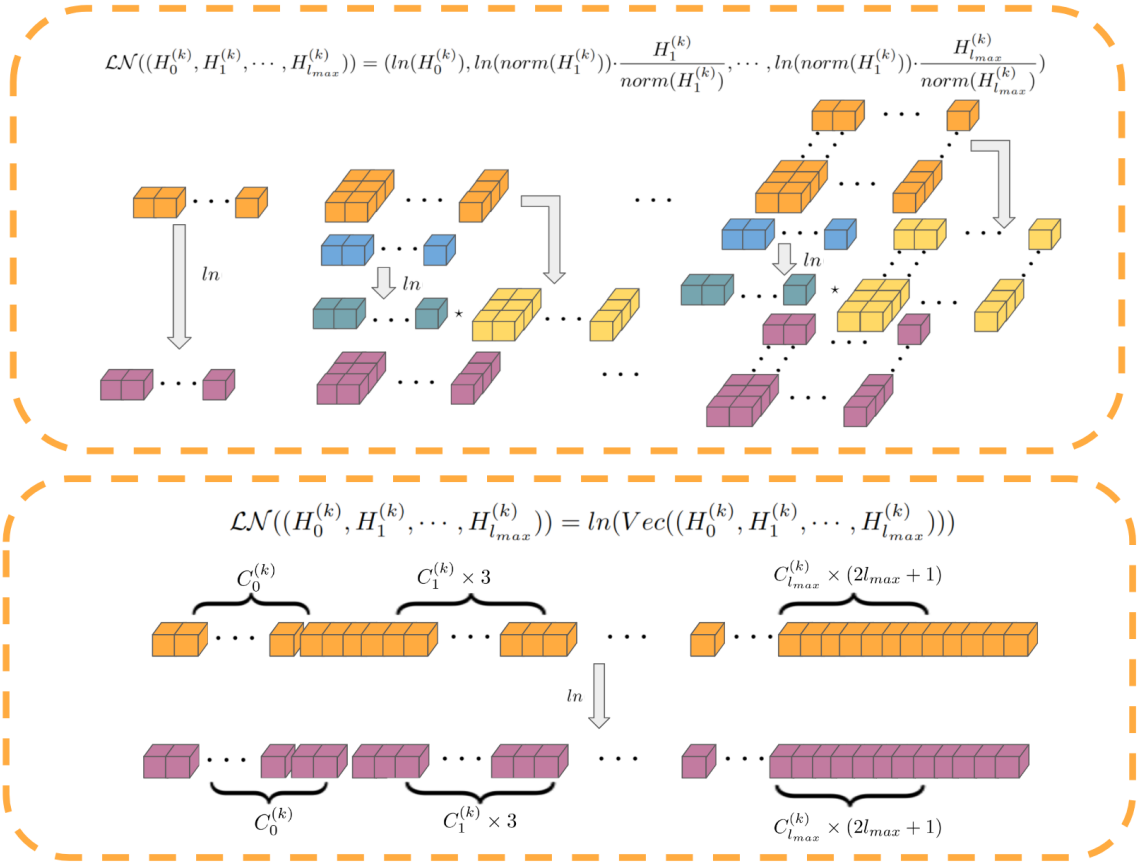


Figure 9: Comparison Between Equivariant and Non-Equivariant Layer Normalization: The figure above illustrates the equivariant nonlinear layer, where layer normalization is applied to the norm of each feature type, followed by multiplication with a unit for each feature. The figure below depicts the traditional nonlinear layer, employing element-wise layer normalization across features, thereby disrupting equivariance by treating each feature type as a concatenation of individual channels rather than a unified geometric entity.

$$
\begin{aligned}
&\mathcal{LN}((H_0^{(k)}, H_1^{(k)}, \cdots, H_{l_{max}}^{(k)})) \\
&= (ln(H_0^{(k)}), ln(\|H_1^{(k)}\|) \cdot \frac{H_1^{(k)}}{\|H_1^{(k)}\|}, \\
&\cdots, ln(\|H_1^{(k)}\|) \cdot \frac{H_{l_{max}}^{(k)}}{\|H_{l_{max}}^{(k)}\|}),
\end{aligned}
$$

where ln is the conventional layer normalization. The difference of equivariant layernormalization and conventional layernormalization are depicted in Figure 9.

A.4 Spherical Harmonics

The spherical harmonics constitute a complete set of orthogonal functions, making them an orthonormal basis. Any spherical function $f \in \mathcal{L}^2(\mathbb{S}^2)$ can be expressed as a linear combination of these spherical harmonics. In simpler terms, they serve as the Fourier basis for functions defined on a sphere.

Similar to the varying frequencies of sines and cosines in Fourier series, spherical harmonics are characterized by different degrees (orders), denoted as $l \in \mathbb{N}$. Each degree of spherical harmonics corresponds to a specific pattern or shape on the surface of a sphere. Higher orders (degrees) indicate higher frequencies, resulting in more intricate and complex patterns on the sphere's surface. To apply spherical harmonics in three-dimensional space ($\mathbb{R}^3$), we incorporate a radial component r^l

into the original spherical harmonics of the corresponding degree (order)-l. This method scales the spherical harmonics for three-dimensional applications, extending their utility beyond the sphere $\mathbb{S}^2$. This adaptation scales the spherical harmonics for applications beyond the two-dimensional sphere surface, effectively extending their utility to three-dimensional analyses and applications. Despite this adaptation, these functions retain their fundamental characteristics and are still referred to as spherical harmonics. In this work, we utilize spherical harmonics, incorporating 3D Cartesian coordinates, as positional embeddings in transformer models.

The fascinating properties of spherical harmonics make them crucial in $SO(3)$ group representations, allowing us to harness their power to achieve equivariance in transformers. A order-l spherical harmonics, denoted as $Y^l : \mathbb{R}^3 \to \mathbb{R}^{2l+1}$, is a vector function with $2l+1$ dimension, following the transformation rule:

$$Y^l(Rr) = D^l(R)Y^l(r),$$
$$Y^l(r) = \|r\|^l Y^l(\hat{r}),$$

where R is an arbitrary rotation, $\hat{r} = \frac{r}{\|r\|}$, and $D^l : SO(3) \to \mathbb{R}^{(2l+1)\times(2l+1)}$, is called the Wigner-D matrix, serving as the irreducible representation of $SO(3)$ corresponding to the order l. The Wigner-D matrix are orthogonal matrix, i.e., $D^l(R)D^l(R)^T = I$. To mitigate the impact of the large scaling factor $\|r\|^l$, we explored alternative approaches: one involved substituting the scaling factor $\|r\|^l$ with Gaussian radial basis functions, expressed as $e^{-(\|r\|-l)^2}$. Another approach entailed employing Fourier sine and cosine series to represent $\|r\|$, creating an invariant embedding. This embedding was then combined with the positional encoding of the unit vector $\hat{r}$ using the spherical harmonics for sphere. However, these alternatives did not demonstrate any significant benefits over the use of spherical harmonics adapted for three-dimensional space ($\mathbb{R}^3$), leading us to continue with our initial methodology.

A.4.1 The use of Spherical Harmonics in Equivariant Transformer

We acknowledge that most equivariant transformer works Fuchs et al. [22], Liao and Smidt [33], Liao et al. [34] also uses spherical harmonics in the transformer layers. However, the use of the spherical harmonics is to derive the equivariant kernel basis based on the relitive position for specific geometirc entities. For example, Fuchs et al. [22] first proposes the $SE(3)$ equivariant transformer for point clouds, where spherical harmonics is served as the equivariant kernel as in steerable 3D convolutions. Equiformer[33, 34] is a graph-based architecture that leverages spherical harmonics for the edges, using a depth-wise tensor product to embed it into the node. For this graph embedding (atom + edge-degree) and graph attention, the use of spherical harmonics could be interpreted more as an equivariant kernel basis. The difference here is that Equiformer uses spherical harmonics to integrate edge information into the equivariant kernel, while ours uses spherical harmonics directly in each token.

B Equivariance of Conventional Positional Encoding

The conventional positional encoding leveraging Fourier sine and cosine functions is translational equivariant. When the input is translated by a translation t, the output will be transformed in a specific way, multiplied by the representation of t, i.e., $PE(x+t) = e^{i\omega t}e^{i\omega x}$ for specific frequency ω in complex format or

$$\begin{aligned} PE(x+t) &= \begin{bmatrix} cos(\omega t) & -sin(\omega t) \\ sin(\omega t) & cos(\omega t) \end{bmatrix} \begin{bmatrix} cos(\omega x) & -sin(\omega x) \\ sin(\omega x) & cos(\omega x) \end{bmatrix} \\ &= \rho_\omega(t) PE(x) \end{aligned}$$

in matrix format.

C Equivariant Hidden Feature Format

The equivariant latent code is structured as $H = \bigoplus_{l \in L} H_l = (H_0, H_1, \cdots, H_l)$, with each H_l having dimensions $(2l+1, C_l)$. The action of a rotation R on this latent code is depicted in Figure 10.

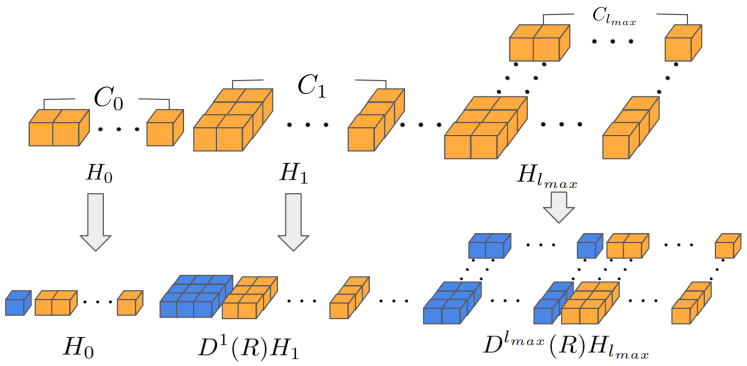


Figure 10: Latent code transformation.

D Visualization of the Equivariant Latent

With the spherical harmonics, we can introduce the Fourier Transform for the sphere. The Fourier coefficient $\mathcal{F}^l$ of a function on the sphere, $f : \mathbb{S}^2 \to \mathbb{R}$, corresponding to the order l is obtained by

$$\mathcal{F}^l = \int_{\mathbb{S}^2} f(x) Y^l(x) dx,$$

and the inverse Fourier Transform without normalization follows the equation:

$$f(x) = \sum_l (\mathcal{F}^l)^T Y^l(x) dx,$$

When we rotate the function f with any rotation $R \in SO(3)$, i.e., we get a new function $f' = f(R^{-1}x)$. The Fourier coefficients corresponding to the order l are:

$$\mathcal{F}'^l = \int_{\mathbb{S}^2} f(R^{-1}x) Y^l(x) dx = \int_{\mathbb{S}^2} f(y) Y^l(Ry) dy = \int_{\mathbb{S}^2} f(y) D^l(R) Y^l(y) dy = D^l(R) \mathcal{F}^l,$$

This indicates that when a spherical function is rotated by any rotation R, its Fourier coefficients will be multiplied by the corresponding Wigner-D matrix. Inversely, we have that when all Fourier coefficients $\mathcal{F}^l$ are multiplied by Wigner-D matrices $D^l(R)$, the obtained spherical function is rotated by R.

With such preliminary, we can treat our latent code $\bigoplus_{l \in L} H_l$ as the Fourier coefficients, where type-l features are the l-th order coefficients, enabling us to derive the spherical function. As a result, when our features are transformed — with each type being multiplied by its respective Wigner-D matrix — the spherical function undergoes a corresponding rotation.

E Equivariant Nonlinear Layer

In our proposed nonlinear layer, we first generate intermediate hidden features $\bigoplus_{l \in L} H'^k_l = (H'^{(k)}_1, \cdots, H'^{(k)}_{l_{max}})$ of the same size as the input via an equivariant linear layer. Subsequent to this, we employ the specified nonlinearity:

$$\begin{aligned}
&\mathcal{A}((H^{(k)}_0, H^{(k)}_1, \cdots, H^{(k)}_{l_{max}})) \\
&= (a(H^{(k)}_0), (a(\langle H^{(k)}_1, H'^{(k)}_1 \rangle) - \langle H^{(k)}_1, H'^{(k)}_1 \rangle) \cdot \frac{H'^{(k)}_1}{norm(H'^{(k)}_1)} + H^{(k)}_1, \\
&\cdots, (a(\langle H^{(k)}_{l_{max}}, H'^{(k)}_{l_{max}} \rangle) - \langle H^{(k)}_{l_{max}}, H'^{(k)}_{l_{max}} \rangle) \cdot \frac{H'^{(k)}_{l_{max}}}{norm(H'^{(k)}_{l_{max}})} + H^{(k)}_{l_{max}})
\end{aligned}$$

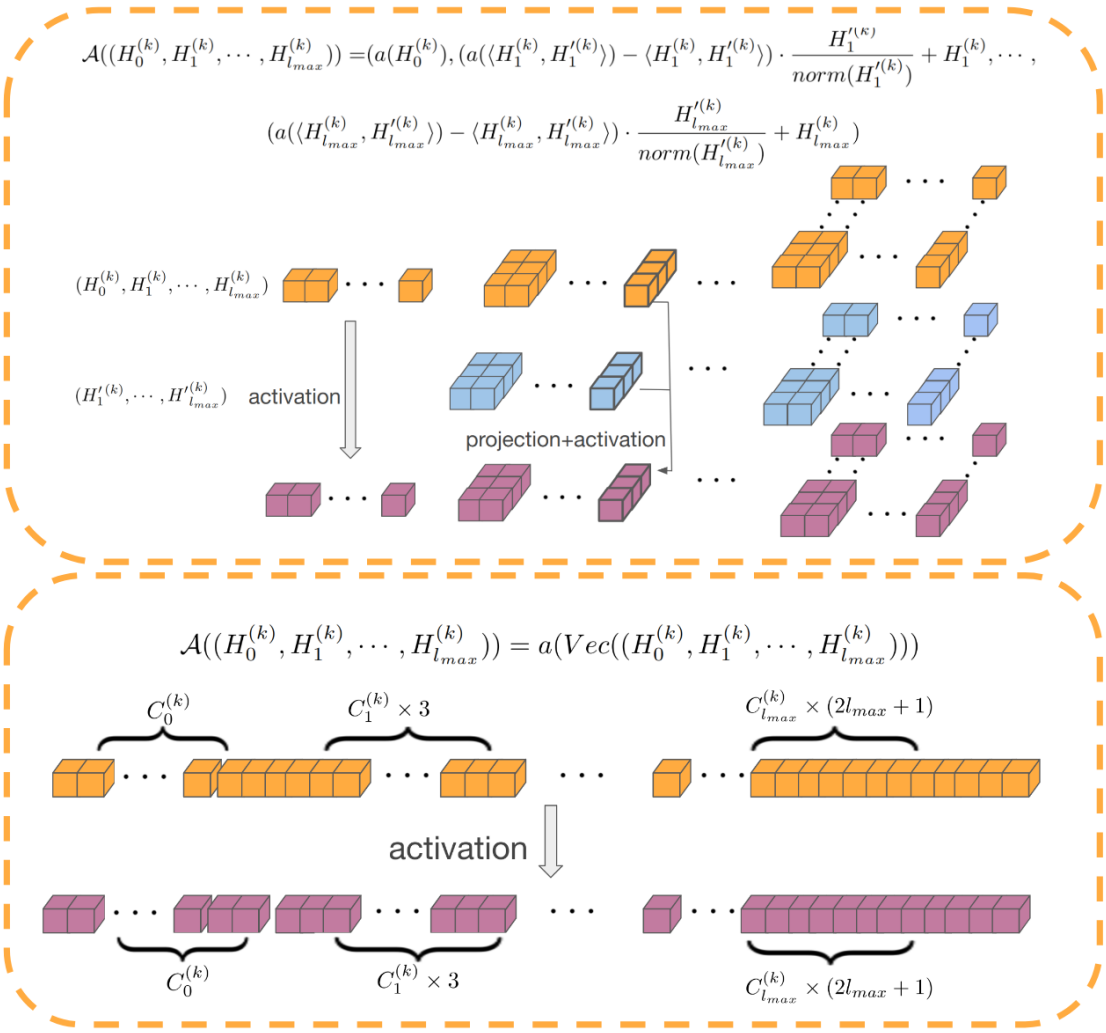


Figure 11: Comparison of Equivariant and Non-Equivariant Nonlinear Layers: The figure above illustrates the equivariant nonlinear layer, similar to those in vector neuron models. This layer establishes equivariant directions using an equivariant linear layer, applies nonlinearity to the projection of the original feature in these directions, and then adds this to the orthogonal component relative to the direction in the input. On the other hand, the figure below shows the conventional nonlinear layer, which applies element-wise nonlinearity to the vectorized feature, thus failing to preserve equivariance.

Here, $\langle \cdot, \cdot \rangle$ denotes the per-channel inner product, meaning the size of $\langle H_l^{(k)}, H_l'^{(k)} \rangle$ is in the format $(1, C_l^{(k)})$. The function a represents a conventional activation operation, such as $ReLU$, $Sigmoid$, or $LeakyReLU$. The symbol $\cdot$ indicates broadcast multiplication. This approach bears resemblance to gated normalization [52]. However, in our model, the scalar for the "gate" is derived from the inner product of two outputs of the equivariant linear layer, rather than being a scalar present in the hidden features themselves. The difference of equivariant nonlinear layers and conventional nonlinear layers are depicted in Figure 11.

F Multi-head Attention Inner Product

Given the input of the attention module formulated as $\bigoplus_{l \in L} H_l$, we generate query Q, key K, and value V by equivariant linear layers. As Q, K, and V are equivariant features, they are expressed as $Q_i = \bigoplus_{l \in L} (Q_l)_i$, $K_j = \bigoplus_{l \in L} (K_l)_j$, and $V_j = \bigoplus_{l \in L} (V_l)_j$, where i and j are the indices of the latents. The inner product is calculated between the equivariant key K and the equivariant query Q, and here we describe how we use it in multi-head attention. With N_h multi-heads, we split the features K, Q, and V into N_h heads along the channel dimension. Taking Q as an instance for clarity,

and denoting C_l as the number of channels for type-l feature $(Q_l)_i$ in Q_i, Q_i gets divided into various heads $(Q_i)^h$ with h as the head index. $(Q_i)^h$ maintains the equivariant feature format, represented as $(Q_i)^h = \bigoplus_{l\in L}(Q_l)_i^h$, with the channel count for type-l feature $(Q_l)_i^h$ being $\frac{C_l}{N_h}$. This division also applies to K and V. The inner product of $(Q_i)^h$ and $(K_j)^h$ is defined as

$$\langle (Q_i)^h, (K_j)^h \rangle = \sum_{l\in L}\sum_{c}^{\frac{C_l}{N_h}} (((Q_l)_i^h)_c)^T ((K_l)_j^h)_c. \tag{1}$$

With the defined inner product, we have the output:

$$(O_i)^h = \sum_j \frac{exp(\langle (Q_i)^h, (K_j)^h \rangle)}{\sum_j exp(\langle (Q_i)^h, (K_j)^h \rangle)} (V_j)^h \tag{2}$$

The final output of the attention mechanism, composed in the channel dimension, can be denoted as $\bigoplus_{l\in L} O_l$. For proof of equivariance, refer to Sec. G.

G Proof of Equivariance for Multi-Head Attention

The inner product is formulated as follows:

$$\langle (Q_i)^h, (K_j)^h \rangle = \sum_{l\in L}\sum_{c=1}^{\frac{C_l}{N_h}} (((Q_l)_i^h)_c)^T ((K_l)_j^h)_c.$$

Owing to the equivariance of the Linear Layer, when the input undergoes a rotation R, the components Q, K, V are correspondingly transformed, denoted as $R\cdot Q$, $R\cdot K$, and $R\cdot V$. Under these conditions, the corresponding inner product becomes:

$$\begin{aligned}
\langle (R\cdot Q_i)^h, (R\cdot K_j)^h \rangle &= \sum_{l\in L}\sum_{c}^{\frac{C_l}{N_h}} ((D^l(R)(Q_l)_i^h)_c)^T (D^l(R)(K_l)_j^h)_c \\
&= \sum_{l\in L}\sum_{c}^{\frac{C_l}{N_h}} (((Q_l)_i^h)_c)^T \boldsymbol{D^l(R)^T D^l(R)} ((K_l)_j^h)_c \\
&= \sum_{l\in L}\sum_{c}^{\frac{C_l}{N_h}} (((Q_l)_i^h)_c)^T (K_l)_j^h)_c \\
&= \langle (Q_i)^h, (K_j)^h \rangle,
\end{aligned}$$

which proves the invariance of the defined inner product. Thereby, the output becomes:

$$\begin{aligned}
&\sum_j \frac{exp(\langle (Q_i)^h, (K_j)^h \rangle)}{\sum_j exp(\langle (Q_i)^h, (K_j)^h \rangle)} (R\cdot V_j^h) \\
&= \sum_j \frac{exp(\langle (Q_i)^h, (K_j)^h \rangle)}{\sum_j exp(\langle (Q_i)^h, (K_j)^h \rangle)} \bigoplus_{l\in L} D^l(R)(V_l)_j^h \\
&= \bigoplus_{l\in L} D^l(R)(\sum_j \frac{exp(\langle (Q_i)^h, (K_j)^h \rangle)}{\sum_j exp(\langle (Q_i)^h, (K_j)^h \rangle)} (V_l)_j^h) \\
&= \bigoplus_{l\in L} D^l(R)(O_l)_j^h \\
&= R\cdot O_j^h,
\end{aligned}$$

which proves that the whole attention mechanism is equivariant.

H Alternative Equivariant Attention

When it comes to attention mechanisms involving latents with different types of equivariant features, a direct method is to use tensor product to entangle these different types, which is complicated and computationally expensive. However, there is an alternative approach that treats the different types of features as Fourier coefficients to obtain spherical features. By applying the conventional transformer to these spherical features, followed by the Fourier Transform, we can retrieve different types of equivariant features. This method offers a more efficient way to entangle and handle various types of features within the attention mechanism. For latent features $\bigoplus_{l\in L} H_l$, we apply the Inverse Fourier Transform so that:

$$S_l(x) = Y^l(x)^T H_l,$$

which implies that S_l has size (N_R, C_l, N_S), where N_R is the number of latents and N_S is the number of samplings for the sphere. From the preliminary in appendix D, we know that when the input features are rotated by R, the output becomes $S_l(R^{-1}x)$, which means these spheres are rotated as well. By concatenating the $\{S_l\}$ on the channel dimension, we get the features after an inverse Fourier Transform with size $(N_R, \sum_l C_l, N_S)$, i.e., we have N_R spheres with $\sum_l C_l$ channels and N_S number of sampling. We can directly apply the self-attention mechanism to these spheres without breaking equivariance, resulting in spherical features F with dimensions $(N_R, \sum_l C_l, N_S)$ after the self-attention. Finally, we apply the Fourier Transform as follows:

$$H_l = \sum_i S^l(x_i) Y^l(x_i),$$

where $S^l(x_i) = F[:, Ind_l, i]$, with Ind_l representing the index of channels for spheres that correspond to type-l features H_l and i denoting the index of the sample on the sphere. By composing different types of features, we obtain outputs in the format $\bigoplus_{l\in L} H_l$. It is evident that the composition of inverse Fourier Transform, transformer on the sphere and the Fourier Transform is equivariant, as confirmed by the preliminary properties of the Fourier Transform in appendix D. In practice, the computational load is increased due to the number of samples and the complexity of spherical convolution. To address this, we can utilize icosahedron sampling and apply equivariant correlation on the icosahedron for the linear layer in attention. Additionally, we can use standard nonlinear layers and typical layer normalization in the self-attention mechanism.

I Averaged Global Geometric Embedding

The formula for the averaged global geometric embedding is as follows:

$$\begin{aligned}
&(\frac{1}{N}\sum_i Y^1(R_i^1) \oplus Y^1(R_i^2) \oplus Y^1(R_i^3),\\
&\frac{1}{N}\sum_i Y^2(R_i^1) \oplus Y^2(R_i^2) \oplus Y^2(R_i^3), \cdots,\\
&\frac{1}{N}\sum_i Y^{l_{max}}(R_i^1) \oplus Y^{l_{max}}(R_i^2) \oplus Y^{l_{max}}(R_i^3)),
\end{aligned}$$

where the superscript denotes the index of the column in the matrix.

J Proof of Invariant Latent and Query

Given that the predicted frame R is equivariant and the latent code $\bigoplus_{l\in L}(\mathcal{R}_K)_l$, is also equivariant, when the input undergoes a transformation by a rotation $(R_0, t_0) \in SE(3)$, the frame is modified to R_0R, and the latent code transforms to $R_0 \cdot \mathcal{R}_K = \bigoplus_{l\in L} D^l(R_0)(\mathcal{R}_K)_l$. Applying the inverse of the equivariant frame to the latent code yields:

$$\bigoplus_{l\in L} D^l(R_0R)^T D^l(R_0)(\mathcal{R}_K)_l = \bigoplus_{l\in L} D^l(R)^T \boldsymbol{D^l(R_0)^T D^l(R_0)} (\mathcal{R}_K)_l = \bigoplus_{l\in L} D^l(R)^T (\mathcal{R}_K)_l,$$

demonstrating that the transformed latent code is invariant. Furthermore, when the input is subjected to a transformation by a rotation $(R_0, t_0) \in SE(3)$, the decoded camera j's pose shifts to $(R_0R_j, R_0t_j +$

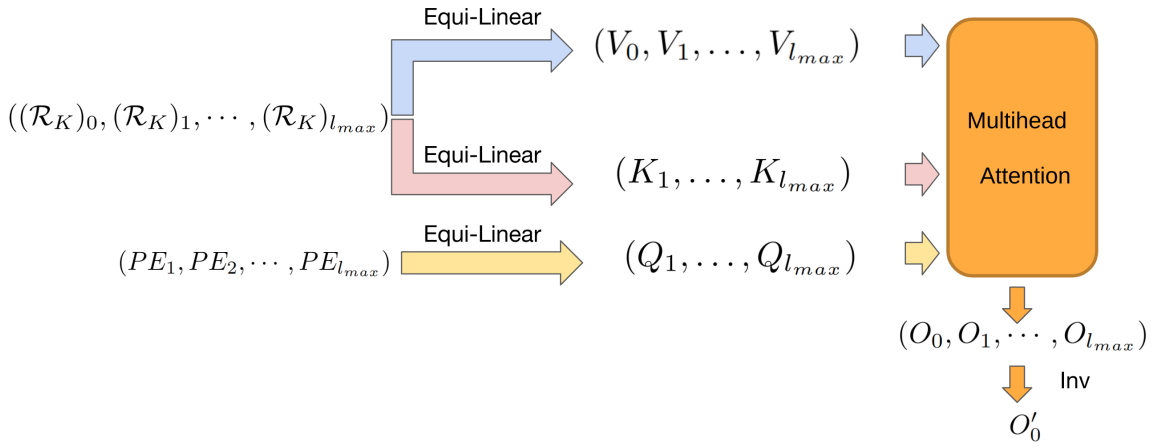


Figure 12: Equivariant Decoder.

t_0). After subtracting the center, the pose becomes $(R_0R_j, R_0(t_j - \bar{t}))$. Applying the inverse of the equivariant frame to the query camera pose, we arrive at:

$$((R_0R)^T(R_0R_j), (R_0R)^T(R_0(t_j - \bar{t}))) = (R^TR_j, R^T(t_j - \bar{t})),$$

which confirms that the transformed query camera pose remains invariant.

K Alternative Equivariant Decoder

The pipeline of the equivariant decoder is show in Figure 12. The cross-attention mechanism with equivariance processes two inputs: firstly, the resulting hidden features from self-attention, denoted as $\bigoplus_{l\in L}(\mathcal{R}_K)_l$ with $L = \{0, 1, \cdots, l_{max}\}$, and secondly, the positional encoding of query rays and cameras, represented as $(PE(r^j_{uv}, t_j - \bar{t}))$ with spherical harmonics. In this context, r^j_{uv} signifies the (u, v)-th ray in the j-th query camera, t_j refers to the translation of the j-th query camera, and $\bar{t}$ is the pre-calculated center of the encoded cameras. The positional encoding is structured as $\bigoplus_{l\in L} PE_l$ with $L = \{1, \cdots, l_{max}\}$. Note that this encoding does not include the 0-type (invariant) image features, in contrast to the encoded input. Hence, when generating the Q, K, V features in the attention module, Q features do not include 0-th type features. To achieve invariant attention weights, K features should also exclude 0-th type features, which is accomplished by setting $W_0 = \mathbf{0}$ in the equivariant linear layer. The multi-head attention mechanism adheres to the equations in Sec. F:

$$\langle (Q_i)^h, (K_j)^h \rangle = \sum_{l\in L} \sum_{c}^{\frac{C_l}{N_h}} (((Q_l)^h_i)_c)^T (((K_l)^h_j)_c),$$

$$(O_i)^h = \sum_j \frac{exp(\langle (Q_i)^h, (K_j)^h \rangle)}{\sum_j exp(\langle (Q_i)^h, (K_j)^h \rangle)} (V_j)^h,$$

where $L = \{1, 2, \cdots, l_{max}\}$, and $(V_j)^h$ conforms to the $\bigoplus_{l\in\{0,1,\cdots,l_{max}\}}(V_l)^h_j$ format. The output O is derived as $\bigoplus_{l\in\{0,1,\cdots,l_{max}\}} O_l$.

Since the final prediction value is unchanged when the reference frame is transformed, it is characterized as an invariant type-0 (scalar) feature. To extract the invariant features for prediction, we utilize an "invariant layer", as depicted in Figure 13. In this process, we initially generate two intermediate features, H' and H'', of the same size. Subsequently, for each type-l, we perform an inner product operation between H'_l and H''_l, which results in invariant features I_l, each with a channel count of C_l. By concatenating these invariant features $\{I_l\}$, we formulate final invariant features $O'_0 = I$ with a combined channel count of $\sum_l C_l$, which is then utilized for the final prediction.

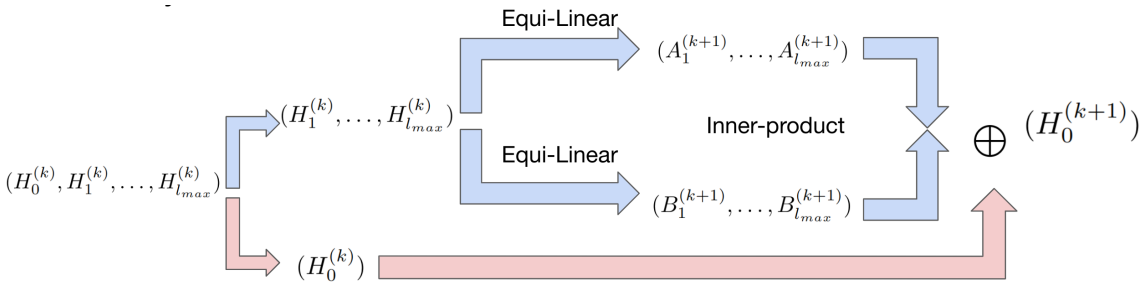


Figure 13: Invariant Layer

Task	Input	Transformation	Pos. Encoding	Feature Embedding	Query	Prediction
Novel View Synthesis	Image + Camera	$SE(3)$	SPH	Image embedding	camera pose	RGB
Neural Volume Rendering	Image + Camera	$SE(3)$	SPH	Image embedding	point+ray dir	σ, RGB
Pose Estimation	Image + Camera	$SE(3)$	SPH	Image embedding	image (Inv)	R, t
Implicit Field for PC	Point	$SE(3)$	SPH	Point embedding/-	point	Field value
$2D$ Dense Prediction	Image + Pixel	SE(2)	Trig	$SO(2)$-equi feature	pixel	Field value

Table 4: Different tasks and their corresponding geometric information in Equivariant Perceiver IO

L Network Architecture and Implementation Details

Regarding architecture, we use a ResNet18 as the visual backbone, resulting in 960-dimensional features. The order of spherical harmonics is [1,2,4,8], resulting in (3+5+9+17)*2 = 68-dimensional features. For encoding, visual and geometric features are concatenated to produce 960 + 68 = 1028-dimensional embeddings. For decoding, we use the same Fourier encoding as the standard Perceiver IO, resulting in 186-dimensional embeddings. Our original latent representation R is of dimensionality 1024 x 512. We set the number of channels for each type of the equivariant hidden feature as [512, 64,32, 8]. In the Perceiver IO implementation, we have 1 block of cross-attention with 1 head, 8 self-attention layers with 8 heads, and 1 cross-attention with 1 head for a decoder.

Our DeFiNe baseline has 73M parameters, and our EPIO implementation has 147M parameters. This increase is due to: additional parameters for the global geometric latent code as shown in Figure 5; inference for frame prediction as shown in Figure 2; and additional parameters for type-2, type-3, and type-4 features, where we set the channel numbers as 64, 32, and 8 respectively. Regarding runtime, we observed an increase of roughly 2x in training iteration times and 1.5x in per-pixel queries during inference. However, we would like to note that our approach converges in roughly 15%. Training and evaluation was conducted using distributed training (DDP) on 8 A100 GPUs, with 80 GB each.

Regarding experiments, we used Pytorch to implement our Equivariant Perceiver IO and will open-source our code and pre-trained weights upon acceptance. We used a batch size of 192, the AdamW optimizer with $\beta = 0.9$, and $\beta_2 = 0.999$, weight decay of 10^{-4}, and an initial learning rate lr at 2×10^{-4}. . For ScanNet, the training duration was 200 epochs, with the learning rate being reduced by half every 80 epochs; For DeMon datasets, the training duration was 200 epochs, with the learning rate being reduced by half every 80 epochs. We used the same losses as DeFiNe, i.e., the L1-log loss, with a weight of 1.0 for real views and 0.2 for virtual views. Following standard practice, we used images of size 128x192 for ScanNet, and images of size 240x320 for DeMoN, using two images as input, with corresponding intrinsics and extrinsics, and ground-truth depth maps as supervision (see Section 3.1). This is the standard training and evaluation protocol and was used by our baselines as well, ensuring a fair comparison.

M Extended Discussion for General Tasks of Equivariant Periceiver IO

Our model, designed as a general architecture, is adaptable to various tasks. This paper focuses on demonstrating the advantages of integrating equivariance into Perceiver IO for scene representation,

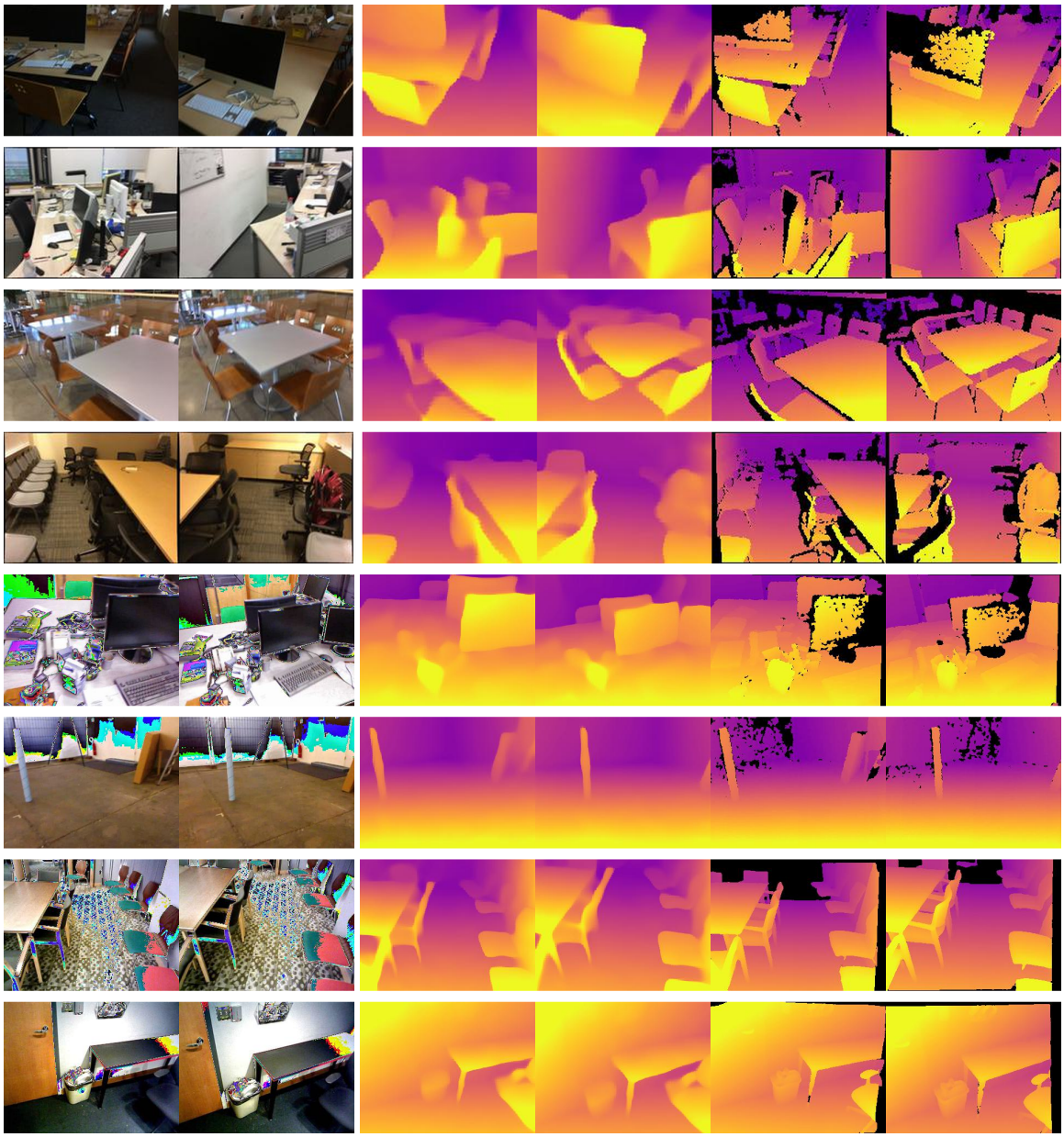


Figure 14: Qualitative Results

primarily evaluated through depth estimation, a core problem in vision. While implementing our model for other tasks is beyond this paper's scope, we provide a brief overview of its potential extensions to different applications, as shown in table 4.

N More Qualitative Results for Depth Estimation

Please see Figure 14 for more qualitative results.

O More Experiments

O.1 Comparison with Current Prevalent Depth Estimation Model

We have evaluated DepthAnything on the same ScanNet stereo benchmark we report results as shown in Table5. we can confidently say that our method outperforms DepthAnything on this benchmark. However, we would like to emphasize that these are not meaningful comparisons. DepthAnything

Models	Abs.Rel.↓	RMSE↓	$\delta < 1.25 \uparrow$
Depth anything	0.099	0.226	0.903
Ours	**0.076**	**0.217**	**0.934**

Table 5: Comparison of our model and Depth Anything

	2 views	3 views	4 views
DeFiNe	0.324	0.315	0.307
Ours	0.215	0.209	0.198

Table 6: Novel View Depth Estimation across a varying number of views.

is a monocular depth estimation network that outputs affine-invariant predictions, while ours is a multi-view depth estimation network that outputs metric predictions. Hence, to achieve the reported DepthAnything numbers, we had to artificially shift and scale predictions using ground-truth depth maps (the same thing is done in their paper). We also could not use the second image as input to DepthAnything, since it is a monocular network, while our method can leverage multiple images as input by design (and even benefits from that during training via the virtual camera augmentation procedure).

O.2 Varying views

We performed additional small-scale experiments to study the impact of the number of available views. In this setting, we have 500 views of one scene, and we randomly choose N encoding views and 1 different decoding camera viewpoint for novel depth estimation.

For DeFiNe, we train with jittering augmentation on the reference frames and test with augmentation as well. For our model, we train without jittering augmentation and also test with augmentation. We explored 2, 3, and 4 views, and the Abs. Rel. depth estimation results are reported in Tab. 6

As we can see, our method consistently surpasses DeFiNe across a varying number of views, even without employing augmentation during training, with the performance gap remaining similar across different view counts.

P Impact Statements

This work aims to advance $3D$ effective learning, with an application in $3D$ reconstruction. While the direct outcome of our research may not have immediate societal implications, the broader application of 3D reconstruction technologies could have some impact. A primary concern is the potential for privacy violations, particularly in scenarios where 3D reconstruction is used to create detailed representations of real-world environments or individuals without their consent. Such applications could lead to unauthorized surveillance or data collection, posing ethical and privacy challenges that need to be addressed as this technology advances and becomes more accessible.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the work in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We don't include propositions or theorems in the paper, but we provide the proofs in the appendix for the statement in the paper for soundness. We provide proof of multi-head attention, invariant latent and query in Appendix. G and Appendix. J.

Guidelines:

- The answer NA means that the paper does our architecture is not limited to specific geometric entities. Safin et al.not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide network architecture and implementation details in Appendix L.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release the code soon.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the training and test details in Section. 4.1 and Appendix. L.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: It was not possible to train independent models enough times to produce enough samples for a statistical analysis. We provide ablations showing the improvements generated by our contributions (all starting from the same random seed), and comparisons to state-of-the-art baselines (which also do not report error bars).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the detials of compute resources in the Appendix. L.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discuss both potential positive societal impacts and negative societal impacts of the work performed in Appendix. P.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the original papers that produce the code package or dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines: The paper does not release new assets.

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines: The paper does not involve crowdsourcing nor research with human subjects.

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.