Ada-VNNs: Adaptive Equivariance for Vector Neural Networks

Bing Han, Ruitao Pan, Yumin Chen, Peixin Hong, Weiyuan Liu, Chenxi Wang, Zhibin Zhao*, and Zhi Zhai

Xi'an Jiaotong University, Xi'an, China.
hbb16@stu.xjtu.edu.cn; zhaozhibin@xjtu.edu.cn
https://github.com/xjtbinghan/Ada_VNNs.

Abstract. Vector Neuron Networks (VNNs) are widely used in 3D tasks for their data efficiency and strong generalization from equivariance. However, their rigid equivariance constraints hinder handling symmetry-breaking, where low-symmetry outputs must be inferred from highly symmetric inputs. In this paper, we reveal the representation collapse issue in VNNs and propose Ada-VNNs, a vector-neuron architecture that can adaptively relax equivariance constraints according to the symmetry level of the data. Ada-VNNs introduce an Adaptive Gating Unit (AGU) to control equivariance relaxation in linear layers, and design a residual pathway that transforms hard structural constraints into adjustable soft priors. We further provide a theoretical characterization that links the learned equivariance behavior to intrinsic symmetry breaking in the data. Experiments on pose estimation across 26 categories with different degrees of symmetry breaking demonstrate that Ada-VNNs enable data-driven adaptive equivariance adjustment, alleviate representation collapse, and significantly improve symmetry-breaking capability over VNN and VN-Transformer, achieving higher performance with almost no additional overhead.

1 Introduction

Symmetry refers to the property of an object or system that remains unchanged or invariant under certain transformations [18, 34]. Incorporating symmetry as an inductive bias in machine learning has emerged as a powerful paradigm, leading to significant conceptual and practical advances [4]. In the 3D domain, VNNs [1,9,20] have become a highly promising class of SO(3)-equivariant models [7, 11, 24, 28, 30, 42], particularly for symmetric geometric and point cloud data. Traditional neural networks require numerous rotated examples to learn such symmetries, whereas VNNs embed them by network structure design, allowing the model to focus directly on learning the intrinsic task mapping.

* Corresponding Author.

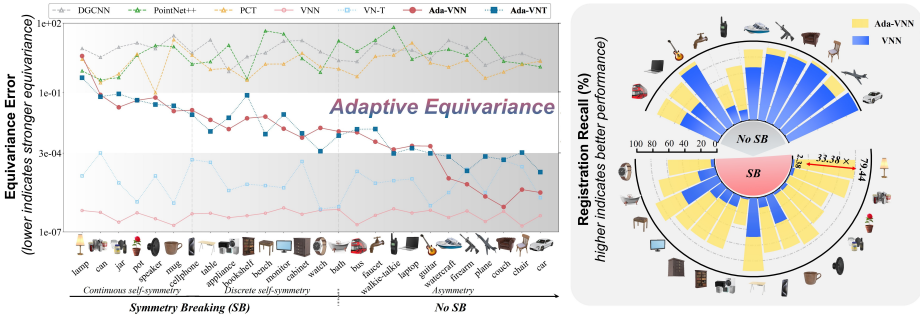

Fig. 1: (Left) Ada-VNN and Ada-VNT can adaptively adjust the model's equivariance constraints according to data distributions with different symmetry levels: on categories with symmetry breaking, the model appropriately relaxes equivariance to improve performance; otherwise, it preserves strong equivariance. (Right) In pose estimation, compared with VNN, our adaptive method handles symmetry-breaking cases more effectively, thereby improving performance on self-symmetric objects.

Equivariant networks are built on symmetry constraints and therefore cannot naturally model symmetry-breaking phenomena [29, 37]. Specifically, when presented with inputs of higher symmetry, they cannot produce outputs of lower symmetry [22]. However, many real-world systems and tasks exhibit spontaneous symmetry breaking. A representative example arises in pose estimation for approximately self-symmetric objects such as cellphones: The overall shape of a cellphone is nearly symmetric, and only weak local cues, such as the camera module or buttons, introduce directional information and determine the final orientation. In this case, the model must exploit such weak asymmetric information to determine a specific orientation from an otherwise highly symmetric input. However, due to their predefined structural constraints, strictly equivariant models tend to exhibit representation dimensional collapse when processing approximately symmetric inputs, which prevents them from effectively exploiting weak asymmetric cues to distinguish orientations.

To address this issue, we propose Ada-VNNs, which can adaptively relax equivariance constraints according to the symmetry level of the data, thereby alleviating representation collapse and enabling effective modeling of symmetry-breaking phenomena under approximately symmetric inputs.

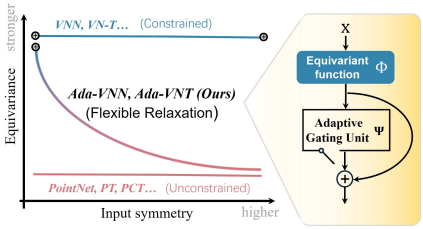

Fig. 2: Ada-VNNs flexibly relax the model's equivariance constraints according to the input's symmetry.

Specifically, we introduce an Adaptive Gating Unit (AGU) that explicitly controls the degree of equivariance relaxation by learning the deviation between weight matrices, turning the linear layers in vector neuron networks from strictly equivariant mappings into adaptively ad-

justable structures. In addition, we design a residual pathway under the VN framework: a strictly equivariant function Φ is connected to the adaptive gated branch Ψ through a residual connection (see Fig. 2). This design preserves the symmetry-induced inductive bias of the original architecture while allowing the model to capture complementary geometric information beyond strict symmetry constraints, effectively transforming hard structural constraints into soft priors. Finally, we provide a theoretical characterization of adaptive equivariance in Ada-VNNs, which establishes a strong correspondence between the model's equivariance behavior and the symmetry level of the input data.

Experiments on pose estimation across 26 categories with different degrees of self-symmetry show that our method achieves adaptive equivariance adjustment according to the symmetry level of the data (see Fig. 1) and effectively alleviates the representation dimensional collapse of strictly equivariant models (see Fig. 5). Compared with VNN and VN-Transformer (VN-T), Ada-VNN and Ada-VNT improve 10° registration recall by 43.2 and 8.33 percentage points, respectively, averaged over the three symmetry regimes (see Tabs. 1 and 2). Notably, these gains come with minimal overhead: the training GPU memory increase of our variants is less than 0.2%.

1. **Problem.** We reveal that strictly equivariant models suffer from representation dimensional collapse when processing approximately self-symmetric inputs, making it difficult for them to exploit weak asymmetric cues for directional discrimination.
2. **Method.** We propose Ada-VNNs, a structure composed of an AGU and a residual pathway, which transforms the hard equivariance constraints of VNNs into adjustable soft priors, thereby enabling the model to adaptively regulate its equivariance constraints.
3. **Theory.** We introduce an adaptive equivariance relaxation theory that theoretically characterizes the correspondence between the degree of equivariance relaxation in Ada-VNNs and the symmetry level of the input data, providing interpretability for the proposed method.

2 Related work

2.1 SO(3)-equivariant Neural Networks

Embedding 3D symmetry into point cloud networks has become a key design principle. Existing SO(3)-equivariant models either approximate continuous rotations with finite subgroups [7,35], use irreducible representations or high-order tensors [14,30], or rely on frame averaging [2]. VNN [9] offers a simpler alternative by replacing scalar features with vector neurons. However, these methods impose rigid predefined equivariance constraints, making them unsuitable for symmetry-breaking tasks.

2.2 Approximately and Relaxed Equivariant Networks

Relaxed equivariance has been explored to address symmetry breaking [10, 19, 25, 32]. Prior works relax weight sharing, allow small equivariance errors, or combine equivariant and non-equivariant branches, as in RPP [13]. Unlike these approaches, we introduce symmetry-aware adaptive relaxation within the VN framework, so the relaxation degree is adjusted according to input symmetry.

2.3 Vector Neuron Networks

VNN [9] replaced traditional scalar features with vector neurons and designed core components—e.g., linear layers, nonlinearities, and normalization—to be equivariant under SO(3). Owing to its simplicity and practical effectiveness, VNN has since been extended to SE(3) (SPD-VN [20]), SIM(3) (EFEM [23]), graphs (E-GraphONet [8]), and attention (VN-T [1]), with applications spanning segmentation [23], registration [40], pose estimation [43], and robotics [38, 39]. OAVNN [3] was the first to address symmetry breaking in VNNs, resolving reflection ambiguity via planar symmetry detection, but its scope is limited to mirror symmetry and point cloud segmentation. We propose the first adaptive framework that effectively handles symmetry breaking under SO(3).

3 Representation Dimensional Collapse

Definition 1. *(Symmetry) An input $x \in X$ is called symmetric if its stabilizer subgroup: $Stab_G(x) = \{g \in G \mid \rho(g)x = x\}$ is non-trivial, here $\rho(g)$ denotes the action of g on x. That is, a symmetric input is fixed by multiple group elements, including at least one non-identity element.*

Next, we introduce the following observation first proposed in [29].

Lemma 1. *(Equivariance constraint). Let X and Y be spaces equipped with a group action of G. We can choose an equivariant $f : X \to Y$ such that $f(x) = y$ only if* $\mathrm{Stab}_G(y) \supseteq \mathrm{Stab}_G(x)$.

This implies that the symmetry of the output of an equivariant function cannot be lower than that of the input. The proof of this lemma can be found in [37] Section E.1. Following [37], we can provide a definition of symmetry breaking at the sample level.

Definition 2. *(Symmetry breaking). Let G be a group. A sample with the input x and output y is* symmetry breaking with respect to G *if* $\mathrm{Stab}_G(y) \not\supseteq \mathrm{Stab}_G(x)$.

Thus, G-equivariant models are inherently incapable of handling symmetry breaking samples with respect to G .

Representation collapse. For strictly equivariant methods, feature collapse occurs when input data possesses approximate symmetry (e.g., objects such as cans or lamps). This strong equivariance constraint leads to performance

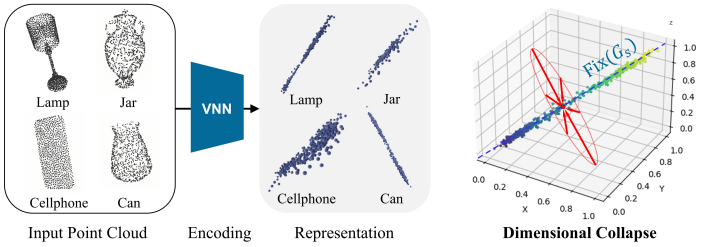

Fig. 3: Representation dimensional collapse phenomenon. When processing self-symmetric inputs, VNN's features collapse to the fixed-point set, leading to the loss of directional information. We provide more examples in Appendix Sec. A.

degradation in tasks like rotation registration. Assume the input x is approximately symmetric with respect to the group G_s; that is, for any $g_s \in G_s$, we have $\rho(g_s)x \approx x$. We define the deviation as $\delta = \|\rho(g_s)x - x\|$. For a strictly equivariant function f, the condition $f(\rho(g_s)x) = \rho(g_s)f(x)$ must hold. Leveraging k-Lipschitz continuity, we obtain:

$$\|\rho(g_s)f(x) - f(x)\| = \|f(\rho(g_s)x) - f(x)\| \leq k\|\rho(g_s)x - x\| = k\delta \tag{1}$$

The term on the left, $\|\rho(g_s)f(x) - f(x)\| \leq k\delta$, implies that when the input is nearly symmetric ($\delta \to 0$), the output feature $f(x)$ is forced to approach the Fixed Point Set of the group G_s. Consequently, if the group G_s represents continuous rotational symmetry about a specific axis, the components of $f(x)$ lying in the plane orthogonal to the rotation axis are forced to approach zero. This results in the loss of critical geometric information required to distinguish orientation, as visualized in Figure 3.

4 Method

To address the above issues, we propose Ada-VNNs, which transform strictly equivariant VNNs (e.g., VNN and VN-T) into adaptive variants that can adjust equivariance constraints according to the symmetry of the input. In this section, we introduce the AGU (Sec. 4.1), the residual structure (Sec. 4.2), and the theoretical analysis (Sec. 4.3).

4.1 Adaptive Gating Unit

Standard neural networks are built upon scalar neurons ($z \in \mathbb{R}$). At layer (d), stacking these scalar neurons yields a latent feature vector $z = [z_1, z_2, \ldots, z_{C^{(d)}}]^\top \in \mathbb{R}^{C^{(d)}}$, where $C^{(d)}$ denotes the number of channels at layer (d). In contrast, VNNs [1, 9, 20] are equivariant networks built upon vector neurons, which lift the neuron representation from a scalar ($z \in \mathbb{R}$) to a 3D vector ($v \in \mathbb{R}^3$).

Accordingly, the feature at each layer is represented as $V \in \mathbb{R}^{C_{\text{in}} \times 3}$, where each row corresponds to one vector neuron and C_{in} is the number of input

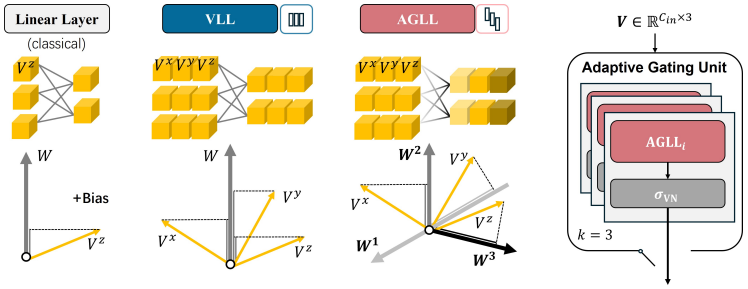

Fig. 4: (Left) Different configurations of linear layer. (Right) The structure of AGU.

channels. By carefully designing the linear, nonlinear, and normalization layers, VNNs achieve strict SO(3)-equivariance.

Specifically, the classical Vector Linear Layer (VLL) in VNN preserves equivariance by removing the bias term and sharing the same weight matrix across the three coordinate channels (see Fig. 4, left). Removing the bias is essential, since a bias introduces a fixed directional component and thus breaks rotational symmetry. Let $V^x, V^y, V^z \in \mathbb{R}^{C_{\text{in}}}$ denote the x, y, and z channel components of V, respectively, and let $W \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$ be the shared weight matrix. Then VLL is defined as

$$\text{VLL}(V) = [WV^x,\ WV^y,\ WV^z]\,, \tag{2}$$

where the output lies in $\mathbb{R}^{C_{\text{out}} \times 3}$. Since all three coordinate channels undergo the same linear transformation, VLL naturally preserves strict equivariance under 3D rotations.

To enable adjustable equivariance, we further propose the Adaptive Gating Linear Layer (AGLL). Unlike VLL, AGLL removes the channel-sharing constraint and applies three independent weight matrices to the three coordinate channels:

$$\text{AGLL}(V) = \left[W^1V^x,\ W^2V^y,\ W^3V^z\right], \tag{3}$$

where $W^1, W^2, W^3 \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$ are three independent weight matrices. When $W^1 = W^2 = W^3$, AGLL reduces exactly to VLL and remains strictly equivariant. In contrast, as the discrepancy among W^1, W^2, and W^3 increases, the transformations across channels become less consistent, and the equivariance constraint is progressively relaxed. Therefore, the distance among these weight matrices can be viewed as an explicit parameterization of the degree of equivariance relaxation. We further analyze the relationship between the weight-matrix discrepancy and the strength of equivariance in Appendix Sec. B.

Finally, we build the AGU (see Fig. 4, right) by using AGLL as the linear layer, VN-ReLU as the nonlinear activation, and stacking three such layers. Its formulation is given as follows:

$$H_1 = \sigma_{\text{VN}}\left(\text{AGLL}_1(V)\right), H_2 = \sigma_{\text{VN}}\left(\text{AGLL}_2(H_1)\right), \text{AGU}(V) = \sigma_{\text{VN}}\left(\text{AGLL}_3(H_2)\right), \tag{4}$$

where AGLL_i denotes the i-th Adaptive Gating Linear Layer and $\sigma_{\mathrm{VN}}(\cdot)$ denotes the VN-ReLU activation. Here, H_1 and H_2 are intermediate hidden features.

4.2 Residual Pathway for Vector Neurons

Inspired by ResNet [17] and RPP [13], we propose a residual architecture for vector-neuron networks:

$$f(x) = \Phi(x) + \Psi(\Phi(x)). \tag{5}$$

The key idea is to decompose the model into two components: a strictly equivariant function Φ, enforced by hard structural constraints, and a flexible corrective function Ψ (see Fig. 2). The branch Φ models the symmetry-consistent part of the data and can be directly instantiated by existing vector-neuron networks, including VN-PointNet, VN-DGCNN, and VN-Transformer, making the framework modular and easy to integrate. The branch Ψ is implemented by the proposed AGU and operates on the equivariant features produced by Φ to compensate for symmetry-breaking information.

This residual design encourages the model to first explain the data using the structured equivariant branch Φ, and to invoke the adaptive branch Ψ only when symmetry breaking is present, in order to correct the part not captured by Φ. In this way, the original hard structural constraints are transformed into adjustable soft priors: the model preserves the symmetry-induced inductive bias while remaining flexible enough to capture complementary geometric information beyond strict equivariance. Moreover, the structured representations generated by Φ provide a more stable and informative input to AGU, which further improves the learning efficiency of the corrective branch.

4.3 Adaptive Equivariance Theory

In this section, we provide a theoretical justification for the adaptive equivariance of the proposed method and reveal the mathematical relationship between the model's equivariance and the symmetry level of the input data. To make this relationship explicit, we first adopt a quantitative formulation of equivariance from [12, 32].

Definition 3. *(Equivariance quantification) A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be C-weakly equivariant with respect to a group G if*

$$\mathcal{E}(f) = \mathbb{E}_{x,g}[\|f(\rho(g)x) - \rho(g)f(x)\|] \leq C,$$

where g is sampled from G according to the normalized Haar measure μ.

A smaller value of $\mathcal{E}(f)$ indicates stronger equivariance of f with respect to G. This definition enables a continuous notion of equivariance strength, and we use equivariance relaxation to refer to allowing a nonzero (or larger) equivariance error compared to a strictly equivariant counterpart when the task itself exhibits symmetry breaking.

We next relate equivariance to task optimization. Let $f : \mathcal{X} \to \mathcal{Y}$ be a neural network mapping from the input space $\mathcal{X}$ to the output space $\mathcal{Y}$, where both spaces are equipped with a group action by G represented by ρ. For supervised equivariant learning, the training objective under group transformations can be written as:

$$\mathcal{L} = \mathbb{E}_{(x,y)\sim\mathcal{D},\, g\sim\mu} \left[\| f\left(\rho(g)x\right) - \rho(g)y \|_2 \right], \tag{6}$$

where g is sampled from G according to the normalized Haar measure μ.

Assumption 1. (Intrinsic symmetry breaking [32]). We assume the data pairs (x, y) are generated by a ground-truth physical process f^*, i.e., $y = f^*(x)$, whose deviation from strict equivariance is bounded by

$$\sup_{x,g} \| f^*(\rho(g)x) - \rho(g)f^*(x) \| = \epsilon, \tag{7}$$

where $\epsilon = 0$ indicates perfect symmetry and $\epsilon > 0$ indicates symmetry breaking.

Proposition 1. (Adaptive equivariance bound). *Under Assumption 1, and assuming the training distribution covers the group orbit (e.g., via data augmentation), the equivariance error of the learned function f is bounded by the task loss and the intrinsic symmetry breaking:*

$$\mathcal{E}(f) \leq 2\mathcal{L} + \epsilon. \tag{8}$$

The detailed derivation is provided in Appendix Sec. C. Equation 8 directly links the achievable equivariance of the learned model to the symmetry level of the underlying task: when the ground truth is (approximately) equivariant (ϵ small), minimizing $\mathcal{L}$ drives $\mathcal{E}(f)$ toward zero; when symmetry breaking is intrinsic ($\epsilon > 0$), a nonzero equivariance error is not only allowed but necessary up to an ϵ-dependent margin. This yields the following two regimes:

Case 1: No symmetry breaking ($\epsilon \to 0$). The bound becomes $\mathcal{E}(f) \leq 2\mathcal{L}$. As training minimizes $\mathcal{L}$, the equivariance error is forced to vanish ($\mathcal{E}(f) \to 0$), and the model converges to a strictly equivariant function.

Case 2: Symmetry breaking ($\epsilon > 0$). The bound becomes $\mathcal{E}(f) \leq 2\mathcal{L} + \epsilon$. Minimizing $\mathcal{L}$ does not enforce zero equivariance error but constrains it within an ϵ-dependent margin, and the model converges to an ϵ-weakly equivariant solution.

Overall, this analysis explains adaptive equivariance: the effective strictness of equivariance is determined by the intrinsic symmetry strength of the data (captured by ϵ), rather than being manually specified.

5 Experiment

Our experiments are designed to answer the following questions: (1) Can Ada-VNNs effectively handle symmetry-breaking problems (Sec. 5.1, 5.2)? (2) Is the proposed adaptive framework compatible with invariant tasks, where invariance is an order-0 special case of equivariance (Sec. 5.3)? (3) Can Ada-VNNs adaptively adjust equivariance according to the symmetry of the input (Sec. 5.4)? (4) What impact do residual connections and adaptive gating have on the model's equivariance and performance (Sec. 5.5)?

Table 1: Comparison results Part 1: Continuous symmetry and Discrete symmetry. Registration recall at a 10° threshold.

Method	Continuous self-symmetry ($\epsilon > 0$)							Discrete self-symmetry ($\epsilon > 0$)								
	lamp	can	pot	jar	speaker	mug	**Avg.**	phone	table	appliance	bookshelf	cabinet	monitor	bench	bath	watch	**Avg.**
PointNet++ [27]	0.00	0.71	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.08
PCT [15]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.30	0.00	0.00	0.00	8.91	7.86	0.00	0.00	2.67
EPN [7]	90.98	70.00	5.98	5.50	84.87	6.20	43.92	83.83	94.49	3.83	30.00	**89.67**	**84.59**	80.31	**93.40**	**93.14**	72.58
VNN [9]	2.38	4.29	1.11	4.51	3.44	37.50	8.87	12.86	49.53	2.33	14.79	15.87	18.72	50.69	22.66	4.12	21.29
Ada-VNN	79.44	75.71	55.56	62.41	76.88	**71.88**	70.31	70.48	93.19	35.58	71.13	70.79	73.52	86.78	73.44	58.76	70.41
Δ	*+77.06*	*+71.42*	*+54.45*	*+57.90*	*+73.44*	*+34.38*	*+61.44*	*+57.62*	*+43.66*	*+33.25*	*+56.34*	*+54.92*	*+54.80*	*+36.09*	*+50.78*	*+54.64*	*+49.12*
VN-T [1]	79.39	53.57	34.44	57.89	75.50	6.25	51.17	76.67	97.33	45.35	71.83	81.30	72.60	**92.84**	78.91	68.04	76.10
Ada-VNT	**92.93**	**77.71**	**57.33**	**67.01**	**85.44**	52.13	**72.09**	**83.52**	**98.21**	**56.33**	**76.20**	87.06	82.34	90.23	80.88	69.67	**80.49**
Δ	*+13.54*	*+24.14*	*+22.89*	*+9.12*	*+9.94*	*+45.88*	*+20.92*	*+6.85*	*+0.88*	*+10.98*	*+4.37*	*+5.76*	*+9.74*	*-2.61*	*+1.97*	*+1.63*	*+4.39*

5.1 Pose Estimation

Pose estimation is a fundamental 3D task. When translation is ignored, relative pose estimation is strictly SO(3)-equivariant. However, many real-world objects exhibit approximate self-symmetry, such as cellphones and lamps, which introduces varying degrees of symmetry-induced ambiguity.

Data and Setting. To systematically evaluate the ability of different models to handle such symmetry breaking, we select 26 object categories from ShapeNet [5] with different levels of approximate self-symmetry. These categories cover three symmetry regimes: categories with approximate continuous self-symmetry, categories with approximate discrete self-symmetry, and categories with little to no self-symmetry. We follow the same experimental setup as VNN and EPN. For each object, we sample a point cloud, apply a random rotation to obtain a transformed version, and pair it with the original point cloud. The network extracts global representations from the two point clouds, and an MLP regresses the relative rotation from these representations. A model that better resolves symmetry-induced ambiguity is expected to learn more direction-sensitive representations, leading to better rotation regression performance.

Baseline and Metric. We compare point cloud networks with different degrees of equivariance: (1) non-equivariant networks, including PointNet++ [27] and PCT [15]; (2) VN-based networks, including VNN [9] and VN-Transformer [1]; and (3) adaptive counterparts, including Ada-VNN and Ada-VNT. Each model is trained separately on the training split of each of the 26 categories. We then evaluate all models on unseen test objects using 10° registration recall, defined as the percentage of predictions whose rotation error is below 10°.

Table 2: Comparison results Part 2: Asymmetry. Registration recall at a 10° threshold.

Method	Asymmetry ($\epsilon \to 0$)											
	bus	laptop	guitar	faucet	walkie-talkie	watercraft	firearm	couch	chair	airplane	car	Avg.
PointNet++ [27]	0.00	1.45	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.13
PCT [15]	0.00	2.10	1.67	3.80	7.61	4.52	0.00	0.00	8.59	10.25	0.00	3.50
EPN [7]	43.52	38.90	88.54	49.91	42.82	**93.16**	85.50	86.10	92.89	94.25	95.22	73.71
VNN [9]	52.86	49.28	89.92	22.52	13.71	66.75	84.84	88.35	92.85	80.72	**97.30**	67.19
Ada-VNN	**85.00**	84.06	94.96	50.45	**70.16**	89.69	**92.84**	91.65	96.46	**96.17**	97.16	**86.24**
Δ	*+32.14*	*+34.78*	*+5.04*	*+27.93*	*+56.45*	*+22.94*	*+8.00*	*+3.30*	*+3.61*	*+15.45*	*-0.14*	*+19.05*
VN-T [1]	72.86	**91.30**	93.96	**72.97**	58.06	86.86	90.74	**93.54**	96.76	93.20	96.87	86.10
Ada-VNT	75.29	88.26	**95.80**	69.46	54.84	88.40	91.45	91.97	**99.19**	93.70	95.16	85.77
Δ	*+2.43*	*-3.04*	*+1.84*	*-3.51*	*-3.22*	*+1.54*	*+0.71*	*-1.57*	*+2.43*	*+0.50*	*-1.71*	*-0.33*

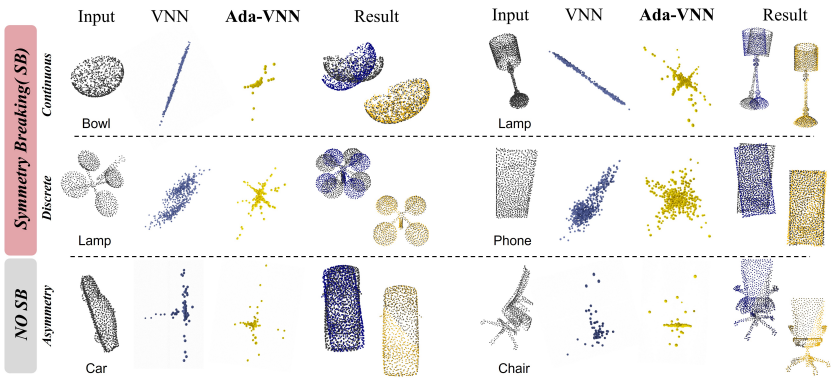

Fig. 5: Representations and pose estimates at different symmetry levels.

Quantitative results. Tab. 1 and Tab. 2 report the quantitative results on categories with different degrees of symmetry breaking. We first observe that the strictly equivariant baselines, VNN and VN-Transformer, suffer substantial performance degradation on categories with symmetry breaking. In particular, VNN nearly fails on categories with strong approximate self-symmetry, especially those with approximate continuous symmetry, indicating that strictly equivariant models struggle to resolve the directional ambiguity introduced by approximately self-symmetric objects, and therefore have difficulty performing accurate pose estimation.

In contrast, our adaptive variants significantly improve the ability of equivariant models to handle symmetry breaking. On categories with approximate continuous self-symmetry, Ada-VNN and Ada-VNT improve performance by approximately 61 and 20 percentage points, respectively. On categories with approximate discrete self-symmetry, Ada-VNN and Ada-VNT further improve performance by 49 and 4 percentage points, respectively. Importantly, these gains do not come at the expense of performance on non-symmetric categories: on categories with little or no self-symmetry, Ada-VNN still outperforms VNN by 19 percentage points, while Ada-VNT maintains performance comparable to VN-Transformer. These results show that our method not only improves robustness to symmetry breaking, but also preserves the effectiveness of equivariant modeling in non-symmetric settings.

In addition, the non-equivariant baselines perform poorly on nearly all categories and generally fail to reduce the rotation error below 10°. We provide a more detailed discussion of this phenomenon in Appendix Sec. D.

Qualitative results.

To further understand this behavior, we visualize the learned representations. As shown in Fig. 5, VNN exhibits clear representation collapse on symmetry-breaking categories, where the features are compressed into a low-dimensional subspace and fail to preserve sufficient direction information. As a consequence, the model cannot reliably infer the correct rotation from such representations.

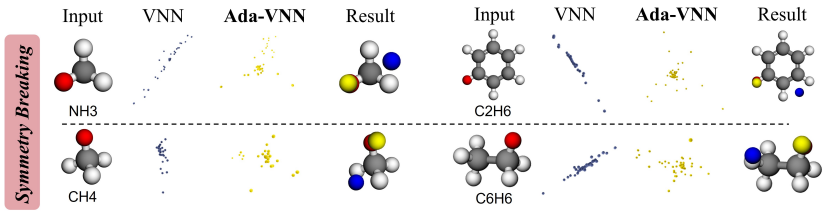

Fig. 6: Representations and isotope predictions at different symmetry levels.

Moreover, we find that the severity of this collapse correlates with the degree of object self-symmetry. Compared with less ambiguous categories such as phone, the learned representations for bowl and lamp, which exhibit stronger approximate continuous self-symmetry, are compressed much more severely, in some cases approaching an almost one-dimensional structure.

By contrast, Ada-VNN and Ada-VNT preserve substantially stronger directional discriminability in their learned representations, especially on bowl and lamp. This suggests that the proposed residual connections and adaptive gating mechanism effectively alleviate representation collapse under symmetry breaking and enhance the model's ability to encode directional information, thereby improving pose estimation performance on approximately symmetric objects. This capability further indicates the potential of our method for a broader range of 3D tasks involving approximate symmetries.

5.2 Isotope Prediction

Symmetry breaking also arises in molecular tasks. For example, isotope substitution in self-symmetric molecules leaves the equilibrium geometry nearly unchanged but changes the atom's physical identity, requiring the model to distinguish originally equivalent sites from an approximately symmetric input, i.e., the input is nearly symmetric while the output must break that symmetry.

Data and Setting. Based on the 3D molecular conformations from QM9 [26], we select four representative molecules with symmetric structures: NH_3, CH_4, C_2H_6, and C_6H_6. For each molecule, we randomly choose one hydrogen atom as the isotope substitution site and apply an extremely small random 3D perturbation only to this site ($\varepsilon \in [10^{-4}, 10^{-3}]$) to construct a controlled weak symmetry-breaking signal. We then apply a random rotation to the entire molecule. For each molecular type, we generate 20,000 training samples and 4,000 test samples. The input to the model is the rotated 3D point set, and the output is the direction vector and radial distance from the molecular reference center to the isotope site.

Baseline and Metric. We use VNN as the equivariant baseline and compare it with Ada-VNN. The evaluation metric is the angular error (in radians) between the predicted and ground-truth directions, which directly measures the model's ability to distinguish directions on approximately self-symmetric targets.

Table 3: Angular error (radians) on isotope prediction. Lower is better.

Method \ Molecule	NH_3	CH_4	C_2H_6	C_6H_6
VNN [9]	0.252	0.631	0.775	0.946
Ada-VNN	0.060	0.155	0.254	0.302
Δ	*-0.192*	*-0.476*	*-0.521*	*-0.644*

Results. As shown in Table 3, Ada-VNN achieves consistently lower directional error than VNN across all molecules, with larger gains on structurally more complex molecules such as C_2H_6 and C_6H_6. As shown in Fig. 6, VNN also exhibits clear representation dimensional collapse on symmetric molecules, whereas Ada-VNN substantially alleviates this issue and preserves stronger directional discriminability. This observation is consistent with our findings in point-cloud pose estimation, and further supports our central claim: in symmetry-breaking tasks, strictly equivariant models tend to lose directional information due to representation collapse, while adaptive equivariance can effectively mitigate this problem.

5.3 Classification

Classification is the order-0 case of our framework, since invariance can be viewed as equivariance with a trivial output representation. Following Sec. 4.3, we use $\rho(g)$ to denote the group action on its argument. Let F denote the final classifier obtained by applying an invariant readout to the backbone representation. Since the class label is unchanged under rotations, the equivariance error in Definition 3 reduces to the invariance error:

$$\mathcal{I}(F) = \mathbb{E}_{x,\, g\sim\mu_G}\left[d\big(F(\rho(g)x), F(x)\big)\right], \tag{9}$$

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_{(x,y)\sim\mathcal{D},\, g\sim\mu_G}\left[d\big(F(\rho(g)x), y\big)\right], \tag{10}$$

$$\epsilon_0 = \sup_{x,\, g\in G} d\big(F^*(\rho(g)x), F^*(x)\big), \tag{11}$$

where G is the augmentation group, $F^*(x) = y$ is the ground-truth label function, and $d(\cdot,\cdot)$ is a metric discrepancy on the classifier output space. Under the same orbit-coverage assumption as Eq. (8), the triangle-inequality argument gives

$$\mathcal{I}(F) \le 2\mathcal{L}_{\text{inv}} + \epsilon_0. \tag{12}$$

For standard object classification, labels are rotation-invariant, so $\epsilon_0 = 0$. Thus, minimizing the group-augmented classification objective encourages the learned classifier to become invariant to the augmentation group. In practice, we train with the standard cross-entropy loss and report the empirical invariance error.

We evaluate point-cloud classification on ModelNet40 [36] under the standard z/z, z/SO(3), and SO(3)/SO(3) train/test settings. Here, z denotes rotations around the upright axis, while SO(3) denotes arbitrary 3D rotations. Eq. (12) applies to the augmentation group used during training: the upright-axis subgroup for z/z, and full 3D rotations for SO(3)/SO(3). The z/SO(3) setting further evaluates out-of-distribution rotation robustness.

Following VNNs [9], we convert the vector-neuron backbone into invariant outputs using an invariant readout. Since Ada-VNNs relax strict equivariance in the backbone, exact invariance is not guaranteed by construction alone. Nevertheless, as shown in Table 4, Ada-VNNs preserve strong rotation robustness. In the challenging z/SO(3) setting, Ada-VND, Ada-VNP, and Ada-VNT substantially outperform the strongest non-equivariant counterparts in their families, improving over DGCNN, PointNet++, and PCT by 53.1, 46.5, and 51.3 percentage points, respectively. Compared with strictly equivariant VN variants, Ada-VNNs show a small accuracy drop in some settings, but their invariance errors remain close to strictly equivariant models and much lower than non-equivariant baselines. These results show that our adaptive framework is also compatible with invariant tasks, where invariance is the trivial-output special case of equivariance.

Table 4: Classification accuracy (%) and empirical invariance error on ModelNet40.

Method	z/ z	z/ SO(3)	SO(3)/ SO(3)	Inv. Err. ↓
DGCNN-based				
DGCNN [33]	90.3	33.8	88.6	0.21
VN-DGCNN [9]	89.5	90.2	89.5	0.0023
Ada-VND	88.3	86.9	87.2	0.0040
PointNet-based				
PointNet [6]	85.9	19.6	74.7	0.20
PointNet++ [27]	91.8	28.4	85.0	0.23
VN-PointNet [9]	77.5	77.5	77.2	0.0035
Ada-VNP	81.2	74.9	77.6	0.0051
Transformer-based				
PT [41]	86.1	24.1	79.4	1.88
PCT [15]	92.8	35.2	89.1	0.82
VN-T [1]	90.0	90.8	89.6	0.0063
Ada-VNT	89.0	86.5	87.4	0.0085

5.4 Adaptive Equivariance

In this section, we aim to verify that the equivariance of Ada-VNNs can adaptively adjust to the degree of symmetry in the input data. Specifically, following the same setup as Sec. 5.1, we train different models on 26 categories and compute their equivariance errors according to Definition 3.

As shown in Fig. 1, non-equivariant models exhibit consistently high equivariance errors ($>10^{-1}$) across all categories, while strictly equivariant models maintain uniformly low equivariance errors ($<3 \times 10^{-4}$) regardless of category. This indicates that neither can adapt their equivariance behavior to the symmetry level of the input data. In contrast, our adaptive models show a markedly different pattern: Ada-VNN and Ada-VNT exhibit larger equivariance errors on categories with stronger symmetry-breaking demands, especially approximately self-symmetric categories, while maintaining lower equivariance errors on categories with little or no self-symmetry. That is, when strict equivariance conflicts with the task objective, the model relaxes the equivariance constraint; otherwise, it preserves the equivariant inductive bias.

We further visualize the evolution of equivariance during training. As shown in Fig. 7, when no symmetry breaking is present, the equivariance error of Ada-VNN gradually decreases throughout training, indicating that the model progressively develops equivariance. In contrast, when symmetry breaking exists, the equivariance error remains difficult to reduce, suggesting that the model tends to preserve non-equivariant behavior. This observation shows that the equivariance of Ada-VNN is not rigidly predetermined by architectural priors,

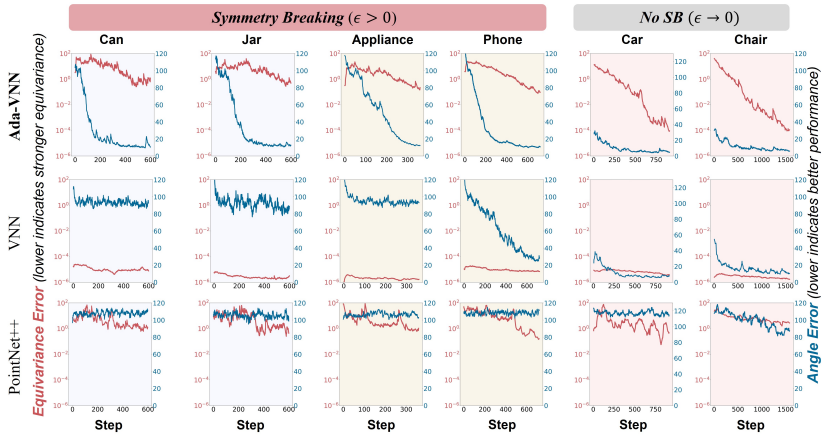

Fig. 7: Evolution of model equivariance and performance during training, where Step denotes the optimization step of each batch.

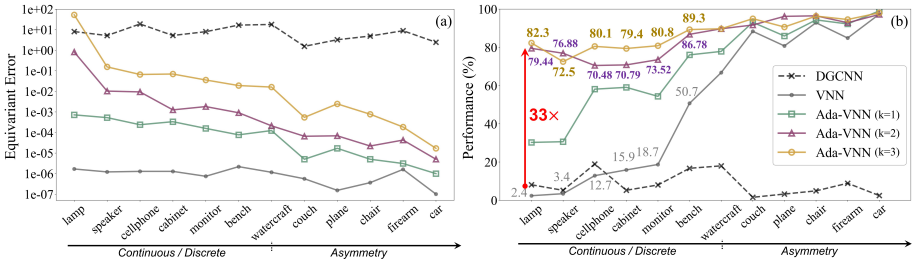

Fig. 8: Effect of AGLL depth k in AGU on (a) equivariance error and (b) pose estimation performance (10° registration recall) across categories.

but is instead formed adaptively during training according to the symmetry structure of the data. More importantly, this also explains the source of its performance gains: when the optimization objective conflicts with strict equivariance constraints, Ada-VNN can improve data fitting by appropriately relaxing its structural equivariance bias, thereby alleviating the mismatch between the model prior and the task objective. These training dynamics further support the conclusion that Ada-VNN outperforms VNN.

5.5 Ablation Study

In this section, we ablate the effects of AGU and residual connections on the equivariance and performance of Ada-VNN.

Effect of AGU. We first study how the number of AGLL layers k in the AGU influences model behavior. Specifically, we set $k \in \{1, 2, 3\}$ and conduct controlled experiments under the same training setup. As shown in Fig. 8, increasing k yields substantial performance gains on symmetry-breaking cate-

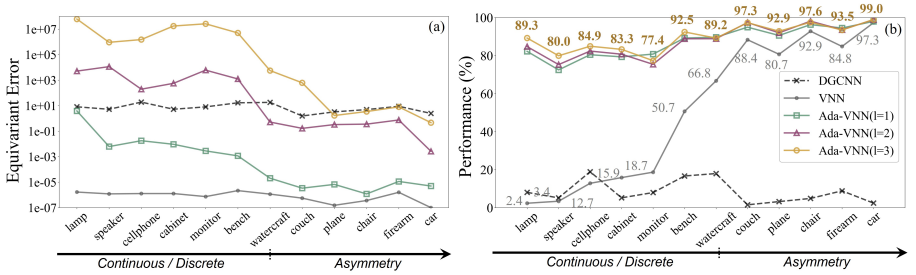

Fig. 9: Effect of residual depth l on (a) equivariance error and (b) pose estimation performance (10° registration recall) across categories.

gories: for example, on the Lamp category, performance improves from 2.4 to ∼30 with $k = 1$, and further to 79.44 with $k = 2$. Meanwhile, the equivariance error increases as k grows: each additional AGLL layer raises the equivariance error by roughly one order of magnitude. This suggests that the capacity of the adaptive (non-strictly equivariant) branch significantly affects the degree of equivariance relaxation, which in turn impacts performance gains under symmetry breaking.

Effect of residual connections. We further ablate the number of residual correction layers in Ada-VNN, setting the residual depth $l \in \{1, 2, 3\}$. As shown in Fig. 9(a), using a single correction layer yields a clear adaptive behavior across categories with different symmetry levels, effectively transitioning from non-equivariant to strongly equivariant regimes. However, increasing the residual depth substantially raises the equivariance error (by roughly four orders of magnitude), suggesting that deeper correction branches introduce stronger equivariance relaxation, consistent with the increased capacity of the non-strictly equivariant components. As shown in Fig. 9(b), a single residual correction layer already brings most of the pose-estimation performance gains, while adding more residual layers yields only marginal improvements. According to the overhead analysis in Appendix Sec. E, one residual layer increases GPU memory usage by only 0.18% yet leads to significant average performance improvements over both VNN and VN-T. These results indicate that Ada-VNN's gains do not come from simply stacking parameters, but from a structural design that directly targets symmetry-breaking scenarios.

6 Conclusion

We identify representation dimensional collapse as a key limitation of strictly equivariant VNNs under symmetry breaking, and propose Ada-VNNs to adaptively relax equivariance through AGU and residual correction. Theoretical analysis links the learned equivariance behavior to intrinsic symmetry breaking, while experiments on pose estimation, isotope prediction, and invariant classification validate the effectiveness and compatibility of the framework. We hope this work provides a step toward more flexible equivariant modeling for 3D tasks.

References

1. Assaad, S., Downey, C., Al-Rfou', R., Nayakanti, N., Sapp, B.: VN-transformer: Rotation-equivariant attention for vector neurons. Transactions on Machine Learning Research (2023)
2. Atzmon, M., Nagano, K., Fidler, S., Khamis, S., Lipman, Y.: Frame averaging for equivariant shape space learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 631–641 (2022)
3. Balachandar, S., Poulenard, A., Deng, C., Guibas, L.: Breaking the symmetry: Resolving symmetry ambiguities in equivariant neural networks. In: NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations (2022)
4. Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P.: Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478 (2021)
5. Chang, A., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. arXiv: Graphics (Dec 2015)
6. Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017). https://doi.org/10.1109/cvpr.2017.16, http://dx.doi.org/10.1109/cvpr.2017.16
7. Chen, H., Liu, S., Chen, W., Li, H., Hill, R.: Equivariant point network for 3d point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14514–14523 (June 2021)
8. Chen, Y., Fernando, B., Bilen, H., Nießner, M., Gavves, E.: 3d equivariant graph implicit functions. In: Computer Vision – ECCV 2022. pp. 485–502. Springer Nature Switzerland (2022)
9. Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., Guibas, L.: Vector neurons: A general framework for so(3)-equivariant networks. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (Oct 2021). https://doi.org/10.1109/iccv48922.2021.01198, http://dx.doi.org/10.1109/iccv48922.2021.01198
10. Elsayed, G., Ramachandran, P., Shlens, J., Kornblith, S.: Revisiting spatial invariance with low-rank local connectivity. In: International Conference on Machine Learning. pp. 2868–2879. PMLR (2020)
11. Esteves, C., Allen-Blanchette, C., Makadia, A., Daniilidis, K.: Learning so(3) equivariant representations with spherical cnns. International Journal of Computer Vision p. 588–600 (Mar 2020). https://doi.org/10.1007/s11263-019-01220-1, http://dx.doi.org/10.1007/s11263-019-01220-1
12. Fei, J., Deng, Z.: Rotation invariance and equivariance in 3d deep learning: a survey. In: Artificial Intelligence Review. vol. 57, p. 168. Springer (2024)
13. Finzi, M., Benton, G., Wilson, A.G.: Residual pathway priors for soft equivariance constraints. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 30037–30049. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/fc394e9935fbd62c8aedc372464e1965-Paper.pdf
14. Fuchs, F., Worrall, D., Fischer, V., Welling, M.: Se(3)-transformers: 3d roto-translation equivariant attention networks. In: Proceedings of the Advances in Neural Information Processing Systems Conference. p. 1970–1981 (2020)

15. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational visual media **7**(2), 187–199 (2021)
16. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
18. Je, J., Liu, J., Yang, G., Deng, B., Cai, S., Wetzstein, G., Litany, O., Guibas, L.: Robust symmetry detection via riemannian langevin dynamics. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
19. Kaba, S., Ravanbakhsh, S.: Symmetry breaking and equivariant neural networks. ArXiv **abs/2312.09016** (2023), https://api.semanticscholar.org/CorpusID:266209980
20. Katzir, O., Lischinski, D., Cohen-Or, D.: Shape-pose disentanglement using se (3)-equivariant vector neurons. In: European Conference on Computer Vision. pp. 468–484. Springer (2022)
21. Kim, S., Park, C., Jeong, Y., Park, J., Cho, M.: Stable and consistent prediction of 3d characteristic orientation via invariant residual learning. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)
22. Lawrence, H., Portilheiro, V., Zhang, Y., Kaba, S.O.: Improving equivariant networks with probabilistic symmetry breaking. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=ZE6lrLvATd
23. Lei, J., Deng, C., Schmeckpeper, K., Guibas, L., Daniilidis, K.: Efem: Equivariant neural field expectation maximization for 3d object segmentation without scene supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4902–4912 (June 2023)
24. Liu, W., Zhu, Y., Zha, Y., Wu, Q., Jian, L., Liu, Z.: Rotation- and permutation-equivariant quantum graph neural network for 3d graph data. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–15 (2025). https://doi.org/10.1109/TPAMI.2025.3593371
25. van der Ouderaa, T., Romero, D.W., van der Wilk, M.: Relaxing equivariance constraints with non-stationary continuous filters. In: Advances in Neural Information Processing Systems. vol. 35, pp. 33818–33830 (2022)
26. Pinheiro, G.A., Mucelini, J., Soares, M.D., Prati, R.C., Da Silva, J.L., Quiles, M.G.: Machine learning prediction of nine molecular properties based on the smiles representation of the qm9 quantum-chemistry dataset. The Journal of Physical Chemistry A **124**(47), 9854–9866 (2020)
27. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems. vol. 30 (2017)
28. Shen, W., Wei, Z., Ren, Q., Zhang, B., Huang, S., Fan, J., Zhang, Q.: Interpretable rotation-equivariant quaternion neural networks for 3d point cloud processing. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(5), 3290–3304 (2024)
29. Smidt, T.E., Geiger, M., Miller, B.K.: Finding symmetry breaking order parameters with euclidean neural networks. Phys. Rev. Res. **3**, L012002 (Jan 2021). https://doi.org/10.1103/PhysRevResearch.3.L012002, https://link.aps.org/doi/10.1103/PhysRevResearch.3.L012002

30. Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., Riley, P.: Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. arXiv: Learning,arXiv: Learning (2018)
31. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
32. Wang, R., Walters, R., Yu, R.: Approximately equivariant networks for imperfectly symmetric dynamics. In: International Conference on Machine Learning. pp. 23078–23091. PMLR (2022)
33. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics p. 1–12 (Oct 2019). https://doi.org/10.1145/3326362, http://dx.doi.org/10.1145/3326362
34. Weyl, H.: Symmetry. Princeton University Press (2015)
35. Worrall, D., Brostow, G.: Cubenet: Equivariance to 3d rotation and translation. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
36. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2015). https://doi.org/10.1109/cvpr.2015.7298801, http://dx.doi.org/10.1109/cvpr.2015.7298801
37. Xie, Y., Smidt, T.: Equivariant symmetry breaking sets. Transactions on Machine Learning Research (2024)
38. Yang, J., Cao, Z.a., Deng, C., Antonova, R., Song, S., Bohg, J.: Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. arXiv preprint arXiv:2407.01479 (2024)
39. Yang, J., Deng, C., Wu, J., Antonova, R., Guibas, L., Bohg, J.: Equivact: Sim (3)-equivariant visuomotor policies beyond rigid object manipulation. In: 2024 IEEE international conference on robotics and automation (ICRA). pp. 9249–9255. IEEE (2024)
40. Yao, R., Du, S., Cui, W., Tang, C., Yang, C.: Pare-net: Position-aware rotation-equivariant networks for robust point cloud registration. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 287–303 (2024)
41. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16259–16268 (October 2021)
42. Zhu, M., Ghaffari, M., Clark, W.A., Peng, H.: E2pn: Efficient se(3)-equivariant point network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1223–1232 (June 2023)
43. Zhu, M., Ghaffari, M., Peng, H.: Correspondence-free point cloud registration with so (3)-equivariant implicit shape representations. In: Conference on robot learning. pp. 1412–1422. PMLR (2022)