Disentangling Rotation and Translation from SE(3)-Equivariant Features for Shape Assembly

Hee-Jun Jung[1], Uigeun Ahn[2], Jin-Hwi Park*[2], and Kangil Kim*[1]

[1] Gwangju Institute of Science and Technology, Gwangju, South Korea
jungheejun93@gm.gist.ac.kr
kangil.kim.01@gmail.com
[2] Chung-Ang University, Seoul, South Korea
ow44459@gmail.com
jinwhipark@cau.ac.kr

Abstract. 3D shape assembly requires predicting 3D pose by estimating rotation and translation separately to align objects or fractures. For 3D pose prediction, prior works utilize the entangled features that capture 3D pose information well, but this mixed representation impedes generalization. To address this problem, we propose the SOT encoder that disentangles 3D pose into an $SO(3)$-equivariant (rotation) feature and a $T(3)$-equivariant (translation) feature. The SOT encoder consists of two branches: 1) a rotation branch that suppresses translation by projecting features onto the translation null space, and 2) a translation branch that suppresses rotation by translation loss and directly subtracts the $SO(3)$-equivariant representation from its $SE(3)$-equivariant counterpart. Across a variety of assembly models and diverse datasets, the SOT encoder consistently demonstrates improved generalization performance in 3D shape assembly. Furthermore, in-depth analyses indicate that the disentangled features show improved equivariant behavior with respect to the target factor while exhibiting invariant behavior with respect to the other. These results indicate that explicitly encouraging factor disentanglement is a straightforward and effective approach for 3D shape assembly. The code is available at https://github.com/maroo-sky/SOT.

1 Introduction

In real-world environments where multiple parts are spatially dispersed, humans intuitively perform shape assembly by inferring which components should connect and estimating the rigid motions required for each, ultimately composing a single and complete object. Representative scenarios include two-object assembly (e.g., inserting a flower into a vase) [28], furniture assembly [13, 19], and further multi-fractured-object reassembly [32]. In all such cases, successful assembly relies on reasoning about the spatial relationships among parts and accurately predicting 3D pose of each component [42], which remains a fundamental challenge in robotic manipulation.

* Co-Corresponding Authors

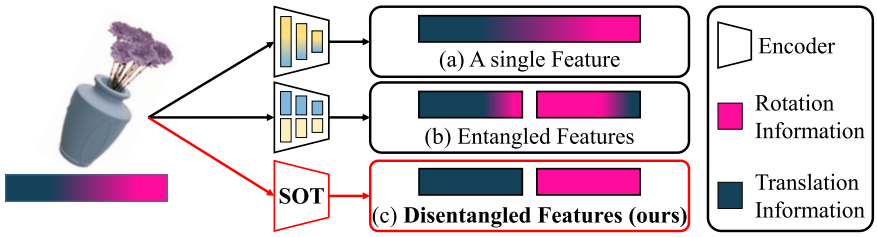


Fig. 1: Illustration of feature disentanglement in 3D shape assembly. While previous methods learn (a) single [7, 17, 27, 28, 42] or (b) entangled [21, 44] features, (c) our proposed encoder explicitly separates rotation (pink) and translation (blue) components, leading to clearer factor-wise representation and better pose generalization.

To estimate the 3D pose of each part, shape assembly methods can be categorized into two approaches: 1) direct pose regression [7, 8, 13, 15, 20, 22, 24, 28, 31, 35, 42, 43, 46] and 2) dense feature matching to establish reliable correspondences [1, 9, 14, 17, 18, 26, 27, 29, 47]. Both approaches jointly predict rotation and translation to recover the whole object. Some works utilize equivariant features [18, 28, 42], as such representations inherently preserve input pose information, thereby improving model generalization. To achieve the equivariant representation, they employ equivariant encoders [10,16,39] that produce a single equivariant feature.

Although such a single $SE(3)$-equivariant feature improves shape-assembly accuracy, it still entangles rotation and translation by encoding both factors in a single representation. This entanglement makes it harder for the model to generalize in 3D pose prediction [21, 44], where the pose naturally decomposes into rotation and translation. For these reasons, disentangled representations, which separate rotation and translation features, tend to support stronger generalization [4, 21, 44]. The disentangled representations address the limitation of the entangled features by encoding factor-wise information independently [4, 34]. Such representations enhance model generalization across diverse computer vision tasks (e.g., classification [2, 45] and object-centric learning [12, 25]), particularly for tasks that require factor-specific reasoning. Building on these approaches, we explore disentanglement in the context of 3D shape assembly to explicitly encourage separation of rotation and translation components, thereby enabling more generalizable 3D pose estimation.

To achieve effective feature disentanglement, we propose an $SO(3)$- and $T(3)$-decomposed encoder (**SOT encoder**) that disentangles rotation- and translation-equivariant features as shown in Fig. 1. Building on dual-branch designs [21, 44] that target rotation/translation decomposition, our method strengthens disentanglement by explicitly suppressing the non-target factor in each branch. We introduce a Rotation-Equivariant and Translation-Invariant Vector-Neuron (RETI-VN) layer that suppresses translations by projecting onto the null space of the translation subspace, while preserving rotational structure. As a result, the output is $SO(3)$-equivariant. To isolate translational information, we obtain a $T(3)$-equivariant feature by translation supervision and subtracting the

$SO(3)$-equivariant representation from its $SE(3)$-equivariant counterpart. These translation supervision and subtraction suppress rotational components while retaining translational variation, yielding a decomposition into rotation and translation features.

By disentangling these two motion factors, the proposed SOT encoder enhances the ability to align parts under arbitrary poses across diverse spatial configurations, leading to improved robustness and generalization in 3D shape assembly. Through quantitative and qualitative experiments, the SOT encoder yields significant improvements on 3D assembly across 2BY2 [28], Breaking Bad [32], and ShapeNet [6] datasets. Our main contributions are:

- We introduce the 1) RETI-VN layer that suppresses translational components, and 2) direct subtraction between the output of $SE(3)$-equivariant function and the RETI function that suppresses rotation information.
- We propose the SOT encoder, which encourages disentangling $SO(3)$- and $T(3)$-equivariant features from arbitrarily transformed inputs, serving as a plug-and-play module for existing frameworks.
- We provide quantitative and qualitative evidence that SOT extracts higher quality $SO(3)$- and $T(3)$-equivariant features than alternative encoders.

2 Related Works

2.1 Shape Assembly

Early works propose the semantic-based methods [13, 23, 41] that utilize the explicit semantic priors (i.e., object labels) from semantically segmented parts. However, these methods depend heavily on semantic annotations and pose challenges for optimizing model performance. Recent geometric-based methods are divided into two categories: 1) direct pose regression [7, 28, 36, 42] and 2) dense feature matching method [3, 17, 18, 27, 38]. The direct pose regression approaches utilize the rotation and translation regressors with a single feature to predict 3D pose information. Dense feature matching builds reliable local correspondences and then performs rigid alignment via hierarchical coarse-to-fine pipelines that exploit global and local geometry [14, 17, 27, 44]. Despite steady progress from both lines of work, prior methods often fail to preserve the rotation and translation information required for accurate 3D pose prediction. In this work, we employ $SE(3)$-equivariant feature representations to achieve robustness under randomly applied $SE(3)$ transformations.

2.2 $SO(3)$- and $SE(3)$-Equivariant Encoders

$SO(3)$- and $SE(3)$-equivariant encoders aim to produce feature representations that remain consistent under 3D transformations. Recent studies learn a single equivariant representation [28, 30, 39, 40, 42], yielding improvements in 3D assembly performance with equivariant [10, 11, 16, 39] architectures. VN Encoder [10] introduces an $SO(3)$-equivariant multi-layer perceptron that operates on vectors rather than scalars, and [11, 39] propose $SE(3)$-equivariant attention mechanisms that linearly combine $SE(3)$-invariant values with $SE(3)$-equivariant

message-passing. These architectures commonly achieve shape assembly with an entangled equivariant feature. Differently, we decompose the $SE(3)$ representations into $SO(3)$- and $T(3)$-equivariant features.

2.3 Decomposition for Rotation and Translation

To decompose rotation and translation information, [16] operates subtraction between two representations to remain rotation information under a strong constraint on the weight, enforcing $\sum_j W_{ij} = 1$ for all i. However, since the constraints on the weight reduce representational capacity [5], the features fail to retain sufficient information. To address this strong constraint, we impose a weaker constraint on the weights. In addition, dual-pipeline approaches [21, 44] decompose rotation and translation using two independent modules rather than learning a single joint representation. However, both methods rely on loss-based separation rather than structural mechanisms. In contrast, we promote disentangling rotation and translation through explicit structural design and provide an in-depth analysis of the resulting representations.

3 Preliminaries

QR Decomposition and Projection Matrix. If $A \in \mathbb{R}^{m\times n}$ has linearly independent columns, its QR factorization $\mathbf{A} = \mathbf{Q}\hat{\mathbf{R}}$ consists of $\mathbf{Q} \in \mathbb{R}^{m\times n}$ with orthonormal columns ($\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_n$) and an upper–triangular $\hat{\mathbf{R}} \in \mathbb{R}^{n\times n}$ with positive diagonal. The columns of $\mathbf{Q}$ form an orthonormal basis for $\mathrm{col}(\mathbf{A})$, and the matrix $\mathbf{P} = \mathbf{Q}\mathbf{Q}^\top$ is the orthogonal projection matrix onto $\mathrm{col}(\mathbf{Q}) = \mathrm{col}(\mathbf{A})$, where $\mathrm{col}(\mathbf{A})$ is a span of column vectors of $\mathbf{A}$.

Group Theory & $SE(3)$-Equivariant Function.

- Group: Let set X, and $(G, \circ)$ be a group, binary operation $\cdot : G \times X \to X$, then group action $\alpha : \alpha(g, x) = g \cdot x$ satisfies following properties: (1) Identity: $e \cdot x = x$, where $e \in G,\ x \in X$, (2) Compatibility: $\forall g_1, g_2 \in G,\ x \in X,\ \alpha((g_1 \circ g_2), x) = \alpha(g_1, \alpha(g_2, x))$.
- G-set: A G-set is a set X, where group action $\phi : G \times X \to X$ is a map such that for all $g \in G$, and $x \in X$.
- Equivariant Function: Given X, Y are G-set, and group action $\rho : G\times Y \to Y$. Then a function $f : X \to Y$ is equivariant if $f(\alpha(g, x)) = \rho(g, f(x))$, where $g \in G, x \in X, y \in Y$.
- $SE(3)$-Equivariant Function: Let G as $SE(3) = \{[\begin{smallmatrix}\mathbf{R} & \mathbf{t}\\ 0 & 1\end{smallmatrix}] \in \mathbb{R}^{4\times4} |\ \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3\}$, where $SO(3)$ is a group of all rotations of 3D Euclidean space, and t is a 3D translation. If $g \in SE(3)$, and $f(\alpha(g, x)) = \rho(g, f(x))$, the function f is $SE(3)$-equivariant function.

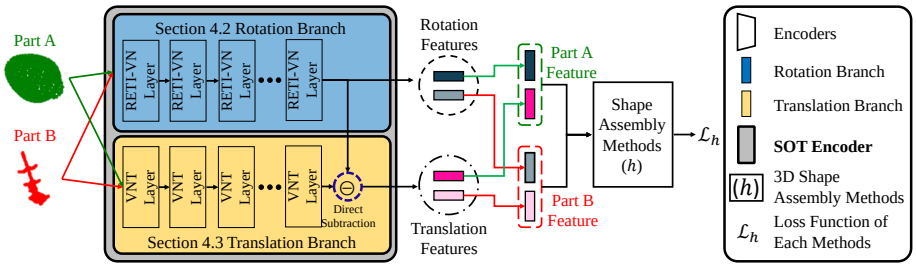


Fig. 2: Overview of SOT encoder. Given part point clouds (left), the SOT encoder disentangles pose information with two branches. The rotation branch (blue; Sec. 4.1) stacks RETI-VN layers to suppress translation and produce an $SO(3)$-equivariant feature. The translation branch (gold; Sec. 4.2) stacks VNT layers to produce an $SE(3)$-equivariant feature, and a direct subtraction ($\ominus$) with the rotation branch and translation supervision suppress rotational components, yielding a $T(3)$-equivariant feature. SOT output features are then passed to a shape-assembly architecture h [7, 18, 28, 42] and optimized with the method-specific loss $\mathcal{L}_h$ [7, 18, 28, 42]. Colors indicate information type: blue/gray for rotation and magenta/pink for translation.

4 Method

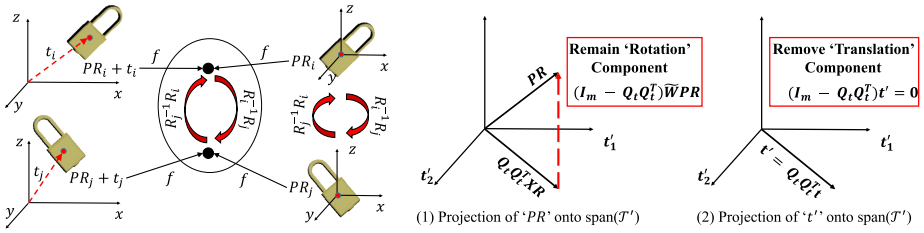
Motivated by dual-branch designs that separate rotation and translation [21, 44], we propose the SOT encoder, which encourages the disentangling of rotation and translation features by reducing the influence of the opposing factor. We introduce the rotation branch that employs a *rotation-equivariant* and *translation-invariant* Vector-Neuron layer (RETI-VN) to suppress translational components, then promoting a rotation representation in Sec. 4.1. We further introduce a translation branch that suppresses rotation-dependent residuals through translation loss and direct subtraction between $SE(3)$- and $SO(3)$-equivariant representations, thereby promoting a translation representation in Sec. 4.2.

4.1 Rotation Branch: Rotation via Translation Suppression

Motivation to Suppress Translation. To construct an $SE(3)$-equivariant layer, conventional method [16] removes translational components and thus yield a rotation feature. Specifically, let VNT weight $\mathbf{W}_i, \mathbf{W}_j$ and the input be $\mathbf{P}' = \mathbf{PR} + \mathbf{1}_n^\top \mathbf{t}$, where $\mathbf{W}_i \in \mathbb{R}^{m\times m}$, $\mathbf{W}_j \in \mathbb{R}^{m\times n}$, $\mathbf{P} \in \mathbb{R}^{n\times 3}$ $\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^{1\times 3}$ and $\mathbf{1}_m = [1, 1, \cdots, 1] \in \mathbb{R}^{1\times m}$. Then it satisfies the following as shown in [16]:

$$\begin{aligned}\mathbf{W}_i\mathbf{W}_j\mathbf{P}' - \mathbf{W}_j\mathbf{P}' &= (\mathbf{W}_i\mathbf{W}_j\mathbf{PR} - \mathbf{W}_j\mathbf{PR}) + (\mathbf{W}_i\mathbf{W}_j\mathbf{1}_n^\top\mathbf{t} - \mathbf{W}_j\mathbf{1}_n^\top\mathbf{t}) \\ &= (\mathbf{W}_i - \mathbf{I})\mathbf{W}_j\mathbf{PR} + (\mathbf{W}_i - \mathbf{I})\mathbf{W}_j\mathbf{1}_n^\top\mathbf{t},\end{aligned} \tag{1}$$

The second term on the right-hand side of Eq. 1 is $\mathbf{0}$ because of the weight constraint of the VNT layer $(\mathbf{W}_i, \mathbf{W}_j)$, that $\sum_l \mathbf{W}_{kl} = 1$ which ensures that $\mathbf{W}_i\mathbf{W}_j\mathbf{1}_n^\top\mathbf{t} = \mathbf{W}_j\mathbf{1}_n^\top\mathbf{t}$. However, strong constraints on the weights limit the model performance [5], and we confirm this empirically in Sec. 5.3. To improve disentanglement of rotation information, we introduce a weakly constrained



(a) RETI function visualization. A rotation-equivariant and translation-invariant function f produces identical features under arbitrary translations, i.e., for any $\mathbf{t} \in \mathbb{R}^{1\times 3}$, $f'(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t}) = f'(\mathbf{PR})$.

(b) Projection visualization. (1) Projection of $\mathbf{PR}$ onto span($\mathcal{T}'$). (2) Projection of $\mathbf{t}' \in$ span($\mathcal{T}'$) onto the same subspace (i.e., onto its own basis $\mathbf{t}'_1, \mathbf{t}'_2$), which leaves $\mathbf{t}'$ unchanged.

Fig. 3: Visualization of the proposed RETI-VN function and the projection operation.

RETI-VN layer that approximately removes translation while preserving rotation equivariance as shown in Fig. 3a.

RETI-VN Layer. Let $\mathbf{W} := (\mathbf{W}_i - \mathbf{I})\,\mathbf{W}_j \in \mathbb{R}^{m\times n}$. Rather than assuming that translation can be eliminated exactly, our goal is to learn a weight matrix that suppresses global translation. Specifically, for an input transformed as $\mathbf{PR}+\mathbf{1}_n^\top \mathbf{t}$, we seek a representation satisfying

$$\mathbf{W}\big(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t}\big) \;\approx\; \mathbf{WPR}, \tag{2}$$

so that translation components are suppressed while rotation variations are preserved as shown in Fig. 3a. To realize this behavior in practice, we construct the RETI-VN layer using a QR decomposition-based projection:

$$\text{RETI-VN}(\mathbf{P}') = \mathbf{WP}' = \mathbf{W}(\mathbf{PR}+\mathbf{1}_n^\top \mathbf{t}) = \mathbf{WPR} + \big(\mathbf{I}_m - \mathbf{Q}_t\mathbf{Q}_t^\top\big)\,\tilde{\mathbf{W}}\mathbf{1}_n^\top \mathbf{t}, \tag{3}$$

where $\mathbf{W} = (\mathbf{I}_m - \mathbf{Q}_t\mathbf{Q}_t^\top)\tilde{\mathbf{W}}$, $\mathbf{I}_m \in \mathbb{R}^{m\times m}$, $\tilde{\mathbf{W}} \in \mathbb{R}^{m\times n}$, and $\mathbf{Q}_t \in \mathbb{R}^{m\times r}$ is obtained by QR decomposition of translation-related directions, such that its columns form an orthonormal basis for the corresponding subspace $\mathcal{T}' := \tilde{\mathbf{W}}\,\mathbf{1}_n^\top \mathcal{T}$. Accordingly, $\mathbf{Q}_t\mathbf{Q}_t^\top$ projects features onto the estimated translation-related subspace, while the complementary projector $\mathbf{I}_m - \mathbf{Q}_t\mathbf{Q}_t^\top$ suppresses components aligned with that subspace. As a result, the translation component in $\tilde{\mathbf{W}}(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t})$ is suppressed after projection, which encourages the model toward Eq. 2 as shown in Fig. 3b.

Objective for Translation Suppression. Since the projection in Eq. 3 alone does not guarantee the target property in Eq. 2, we introduce a regularizer that explicitly suppresses the response of $\mathbf{W}$ to translations as below:

$$\mathcal{L}_w = \frac{1}{i}\sum_i \Big|\sum_j \mathbf{W}_{i,j}\Big|. \tag{4}$$

We regularize the sum of each row of $\mathbf{W}$ to be close to zero, i.e., $\sum_j \mathbf{W}_{i,j} \approx 0$, since $\mathbf{W}\mathbf{1}_n^\top \mathbf{t} = [\sum_j \mathbf{W}_{1,j}\mathbf{t}_1,\ \sum_j \mathbf{W}_{2,j}\mathbf{t}_2,\ \sum_j \mathbf{W}_{3,j}\mathbf{t}_3]$, which shows that the translation response is governed by the row sums of $\mathbf{W}$. Hence, encouraging each row sum to be small makes $\mathbf{W}(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t})$ closer to $\mathbf{WPR}$ in practice.

Learnable Orthonormal Basis $\mathbf{Q}_t$. Constructing the orthonormal basis $\mathbf{Q}_t$, we parameterize the subspace with a learnable matrix $\mathbf{W}_Q \in \mathbb{R}^{m\times r}$ and obtain an orthonormal basis via QR factorization, $\mathbf{W}_Q = \mathbf{Q}_t\hat{\mathbf{R}}$, where $\mathbf{Q}_t \in \mathbb{R}^{m\times r}$ has orthonormal columns and $\hat{\mathbf{R}} \in \mathbb{R}^{r\times r}$ is upper triangular. We then employ $\mathbf{Q}_t$ in Eq. 3 together with the learnable parameter $\tilde{\mathbf{W}}$.

Moreover, since $\mathrm{span}(\mathcal{T}')$ depends on the observed samples, we treat the subspace dimension r in $\mathbf{Q}_t \in \mathbb{R}^{m\times r}$ as a hyperparameter, choosing $1 \le r \le m$ to accommodate dataset-specific variability, where m denotes the output channel dimension. Following VN-DGCNN [10], which stacks nonlinear vector-neuron layers, we stack RETI-VN layers equipped with an $SE(3)$-equivariant nonlinear activation function [16], as shown in the blue modules of Fig. 2.

4.2 Translation Branch: Translation via Rotation Suppression

Translation Supervision Suppresses Rotation-Dependent Residuals. Many shape assembly methods employ an $SO(3)$- or $SE(3)$-equivariant encoder as a feature extractor together with $SE(3)$-equivariant models [16,28,42]. In this setting, translation supervision not only recovers the translational component, but also suppresses the residual term that still depends on rotation. To make this precise, let $g := h \circ f : \mathbb{R}^{n\times 3} \to \mathbb{R}^{n\times 3}$ denote the composite map of an encoder f and a shape assembly $SE(3)$-equivariant model h. If encoder f is an $SE(3)$-equivariant map, then g is $SE(3)$-equivariant under the natural action on point clouds:

$$g(\mathbf{P}\mathbf{R} + \mathbf{1}_n^\top \mathbf{t}) = g(\mathbf{P})\mathbf{R} + \mathbf{1}_n^\top \mathbf{t}, \qquad \mathbf{R} \in SO(3),\ \mathbf{t} \in \mathbb{R}^{1\times 3}. \tag{5}$$

Following prior works [7, 28, 42], we define the translation loss by matching the predicted translation against the ground-truth translation:

$$\mathcal{L}_{\text{trans}} = \left\| g(\mathbf{P}\mathbf{R} + \mathbf{1}_n^\top \mathbf{t}) - \mathbf{1}_n^\top \mathbf{t} \right\|_2 . \tag{6}$$

Proposition 1. *Minimizing $\mathcal{L}_{\text{trans}}$ is equivalent to suppressing the rotation-dependent residual term $g(\mathbf{P})\mathbf{R}$.*

Proof. By Eq. (5),

$$\mathcal{L}_{\text{trans}} = \left\| g(\mathbf{P}\mathbf{R} + \mathbf{1}_n^\top \mathbf{t}) - \mathbf{1}_n^\top \mathbf{t} \right\|_2 = \left\| g(\mathbf{P})\mathbf{R} + \mathbf{1}_n^\top \mathbf{t} - \mathbf{1}_n^\top \mathbf{t} \right\|_2 = \left\| g(\mathbf{P})\mathbf{R} \right\|_2 . \tag{7}$$

Since $\mathbf{R} \in SO(3)$ is orthogonal, the L2 norm is right-invariant under multiplication by $\mathbf{R}$, and thus

$$\left\| g(\mathbf{P})\mathbf{R} \right\|_2 = \left\| g(\mathbf{P}) \right\|_2 . \tag{8}$$

Therefore, $\mathcal{L}_{\text{trans}} = \|g(\mathbf{P})\|_2$, which shows that minimizing the translation loss is exactly equivalent to suppressing the rotation-dependent residual carried by the $SE(3)$-equivariant output. ■

Limitation of the Composite Function of $SE(3)$-Equivariant Maps. Under the idealized zero-loss condition, Eq. 8 gives $g(\mathbf{P})\mathbf{R} = \mathbf{0}$. Since $\mathbf{R} \in SO(3)$ is orthogonal and hence invertible, it follows that $g(\mathbf{P}) = h(f(\mathbf{P})) = \mathbf{0}$.

Proposition 2. *Let $f : \mathbb{R}^{n\times 3} \to \mathbb{R}^{m\times 3},\ h : \mathbb{R}^{m\times 3} \to \mathbb{R}^{n\times 3}$ be $SE(3)$-equivariant maps under the natural point-cloud $\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t},\ (\mathbf{R}, \mathbf{t}) \in SE(3)$. Then the condition $h(f(\mathbf{P})) = \mathbf{0}\ \forall \mathbf{P} \in \mathbb{R}^{n\times 3}$ is impossible.*

Proof. Define $g := h \circ f$. Since both f and h are $SE(3)$-equivariant, g is also $SE(3)$-equivariant: $g(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t}) = g(\mathbf{P})\mathbf{R} + \mathbf{1}_n^\top \mathbf{t}$. Assume $g(\mathbf{P}) = \mathbf{0}$ for all $\mathbf{P}$. Then $\mathbf{0} = g(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t}) = g(\mathbf{P})\mathbf{R} + \mathbf{1}_n^\top \mathbf{t} = \mathbf{1}_n^\top \mathbf{t}$, which is impossible for any $\mathbf{t} \neq \mathbf{0}$. Therefore, $h(f(\mathbf{P})) = \mathbf{0}\ \forall \mathbf{P} \in \mathbb{R}^{n\times 3}$ cannot hold. ■

Discrepancy-Based Direct Subtraction. The above result shows that directly enforcing $h(f(\mathbf{P})) = \mathbf{0}$ is overly restrictive: it requires the composite map $h \circ f$ to behave as a zero map, which is incompatible with the affine $SE(3)$ action induced by translation. Motivated by this limitation, we instead introduce an auxiliary process and optimize the following discrepancy-based objective:

$$\begin{aligned} \mathcal{L}_{trans} &= \left\| h\big(f(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t}) - f'(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t})\big) - \mathbf{1}_n^\top \mathbf{t} \right\|_2 \\ &= \| h(f(\mathbf{P}) - f'(\mathbf{P}))\mathbf{R} \|_2 \end{aligned} \tag{9}$$

where f' is a RETI-VN. Unlike the direct constraint, this objective does not force either $f(\mathbf{P})$ or $h(f(\mathbf{P}))$ itself to vanish. Instead, it encourages the model to reduce the discrepancy between the $SE(3)$-equivariant network f and the RETI network f'. As a result, the proposed discrepancy branch yields a less restrictive training objective than directly enforcing $h(f(\mathbf{P})) = \mathbf{0}$. We empirically support this interpretation through the performance improvements observed in our ablation results in Sec. 5.3 and disentangled representation analysis in Sec. 5.4.

4.3 Objective Loss Implementation

Following prior works, we directly predict rotation and translation. Accordingly, we define the overall objective as

$$\mathcal{L} = \mathcal{L}_h + \mathcal{L}_w, \tag{10}$$

where $\mathcal{L}_h$ is the objective loss of previous shape assembly methods h such as NSM [7], SE(3)-assembly [42], and 2BY2NET [28], which include rotation $\mathcal{L}_{rot}$ and translation loss $\mathcal{L}_{trans}$. When using the SOT, we replace these losses $\mathcal{L}_{rot} = ||h \circ f'(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t}) - \mathbf{R}||_2$, $\mathcal{L}_{trans}$ defined in Eq. 9, and $\mathcal{L}_w$ defined as Eq. 4.

Implementation Details For clarity, the preceding formulation operates at the point-coordinate level, where $\mathbf{W} \in \mathbb{R}^{m\times n}$ acts on the point dimension to suppress translational components in point coordinates. In implementation, SOT follows the VN/VNT framework. In implementation, SOT follows the VN/VNT framework. Given vector-list features $\mathbf{V} \in \mathbb{R}^{N\times C_{\ell-1}\times 3}$ following the VN formulation [10], RETI-VN applies $\mathbf{W}' \in \mathbb{R}^{C_\ell \times C_{\ell-1}}$ along the channel dimension and shares it across points, yielding $\mathbf{V}' = \mathbf{W}'\mathbf{V} \in \mathbb{R}^{N\times C_\ell \times 3}$. The overall procedure is summarized in Algorithm 1. VFeat$(\cdot)$ denotes an initial feature extractor that maps the input point cloud $\mathbf{P}'$ into vector-list features $\mathbf{V}^{(0)} \in \mathbb{R}^{B\times N\times C_0\times 3}$.

Algorithm 1 SOT Forward Pass

Require: $\mathbf{P}' = \mathbf{P}\mathbf{R} + \mathbf{1}_n^\top \mathbf{t}$; RETI-VN params $\{\tilde{\mathbf{W}}'^{(\ell)}, \mathbf{W}_Q'^{(\ell)}\}_{\ell=1}^{L_r}$
Ensure: $\mathcal{L}, \hat{\mathbf{R}}, \hat{\mathbf{t}}$
(0) Feature initialization: $\mathbf{V}^{(0)} \leftarrow \mathrm{VFeat}(\mathbf{P}')$ $\{\mathbf{V}^{(0)} \in \mathbb{R}^{B\times N\times C_0\times 3}\}$
(1) Rotation branch: $\mathcal{L}_w \leftarrow 0$
for $\ell = 1, \ldots, L_r$ **do**
  $\mathbf{Q}_t^{(\ell)} \leftarrow \mathrm{QR}(\mathbf{W}_Q'^{(\ell)})$, $\mathbf{\Pi}^{(\ell)} \leftarrow \mathbf{I} - \mathbf{Q}_t^{(\ell)}\mathbf{Q}_t^{(\ell)\top}$
  $\mathbf{W}'^{(\ell)} \leftarrow \mathbf{\Pi}^{(\ell)}\tilde{\mathbf{W}}'^{(\ell)}$
  $\mathbf{V}_{b,p,:,:}^{(\ell)} \leftarrow \sigma_{\mathrm{VN}}(\mathbf{W}'^{(\ell)}\mathbf{V}_{b,p,:,:}^{(\ell-1)}), \forall b, p$
  $\mathcal{L}_w \mathrel{+}= \frac{1}{C_\ell}\sum_i |\sum_j \mathbf{W}_{ij}'^{(\ell)}|$
end for
$f'(\mathbf{P}') \leftarrow \mathbf{V}^{(\ell)}$
(2) Translation branch: $\mathbf{V}' \leftarrow \mathrm{VNT}(\mathbf{P}')$
$\hat{f}(\mathbf{P}') \leftarrow \mathbf{V}' - f'(\mathbf{P}')$
(3) Joint pose prediction: $(\hat{\mathbf{R}}, \hat{\mathbf{t}}) \leftarrow h(f'(\mathbf{P}'), \hat{f}(\mathbf{P}'))$
(4) Loss: $\mathcal{L} \leftarrow \mathcal{L}_h + \mathcal{L}_w$
return $\mathcal{L}, \hat{\mathbf{R}}, \hat{\mathbf{t}}$

Table 1: Two-part 3D shape assembly results. An ‡ denotes the encoder used in the 3D shape assembly method paper [7, 28, 42].

Methods	Encoder	2BY2 [28]				Breaking Bad (everyday) [32]				ShapeNet [6]			
		RMSE-R ↓	RMSE-T ↓	CD ↓	PA ↑	RMSE-R ↓	RMSE-T ↓	CD ↓	PA ↑	RMSE-R ↓	RMSE-T ↓	CD ↓	PA ↑
NSM [7]	DGCNN‡	37.4801	0.4019	0.7360	0.5488	78.3134	0.1295	0.0781	0.8824	54.9841	0.1238	0.0961	0.8596
	VN-DGCNN	44.3693	0.4853	0.7518	0.5165	85.8648	0.1412	0.0879	0.8655	65.7310	0.1325	0.1198	0.8158
	FINet-EB	37.0824	0.4197	0.5914	0.5664	87.3287	0.1328	0.0979	0.8393	64.9645	0.1297	0.1124	0.8272
	SOT (ours)	**29.1815**	**0.3982**	**0.3490**	**0.6121**	**71.9768**	**0.1100**	**0.0769**	**0.8887**	**44.9236**	**0.1157**	**0.0893**	**0.8694**
SE-3 Assembly [42]	DGCNN	43.1994	0.4403	1.1484	0.4403	75.9024	0.1005	0.1957	0.7425	46.6635	0.1322	0.1410	0.8061
	VN-DGCNN‡	39.1817	0.4113	2.2576	0.5536	53.5764	0.0814	0.1104	0.8423	59.1983	0.1038	0.1497	0.7814
	FINet-EB	36.4079	0.3569	1.4027	0.5243	55.3831	0.0692	0.1087	0.8402	75.7370	0.1086	0.1579	0.7719
	SOT (ours)	**32.3640**	**0.3039**	**0.6200**	**0.5911**	**49.2427**	**0.0684**	**0.0842**	**0.8662**	**37.1323**	**0.0957**	**0.1100**	**0.8383**
2BY2NET [28]	DGCNN	30.4387	0.3386	2.2349	0.5207	72.0284	0.0865	0.1813	0.7575	82.8681	0.1301	0.2872	0.6425
	VN-DGCNN‡	29.7778	0.3597	4.9334	0.5330	59.8457	0.0685	0.1396	0.8181	60.2706	**0.0765**	0.1462	0.7945
	FINet-EB	30.3856	0.4391	6.1800	0.3812	62.0458	0.0724	0.1683	0.7976	62.2055	0.1906	0.1983	0.7829
	SOT (ours)	**22.4719**	**0.2857**	**1.5265**	**0.5623**	**53.7092**	**0.0568**	**0.0873**	**0.8533**	**51.6656**	0.0799	**0.1109**	**0.8179**

5 Experiments

5.1 Environmental Settings

Datasets. To validate our method across diverse settings, we use three assembly datasets that span different scales: ShapeNet [6] as a small-scale benchmark, and Breaking Bad [32] together with 2BY2 [28] as large-scale benchmarks.

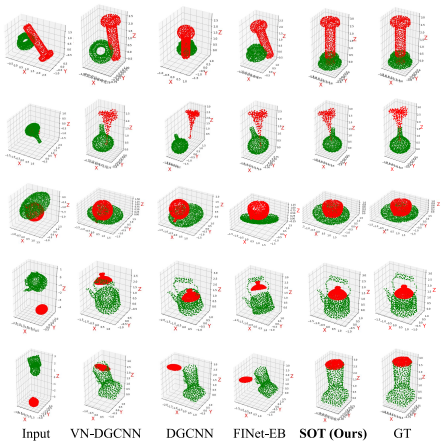
Evaluation Metrics. We assess pose accuracy using the Root Mean Squared Error (RMSE) computed separately for rotation and translation [42]. Specifically, RMSE-R measures the discrepancy between the predicted rotation $\mathbf{R}$ (parameterized by Euler angles) and the ground truth, while RMSE-T measures the discrepancy between the predicted translation $\mathbf{t}$ and the ground truth. Assembly quality is evaluated using the Chamfer Distance (CD) and Part Accuracy (PA). Following prior work [22, 23], a part is counted as correct if its CD to the ground-truth counterpart is below the threshold 0.1.

Wide-Range 3D Translation Setting. Most prior works [27, 28, 33, 38] first apply a random rotation and then translate each part to its centroid, i.e., they fix the translation vector as $t_i = C_{P_i}$, where C_{P_i} is the centroid of part P_i. This choice yields a deterministic (non-random) translation for every part. However, real-world fragments can appear at arbitrary locations, so we allow random translations within a bounded range and sample $t_i \sim \mathcal{U}\big([-2\,|C_{P_i}|,\ 2\,|C_{P_i}|]\big)$, where $\mathcal{U}(\cdot)$ denotes a uniform distribution, and the uniform range is applied independently to each coordinate axis.

Table 2: Additional shape assembly tasks on the Breaking Bad dataset. (a) We apply the SOT encoder to the CMNet [18] method for the two-part assembly task. (b) We experiment on a multi-part assembly task with SE(3)-Assembly.

(a) Dense feature matching method.

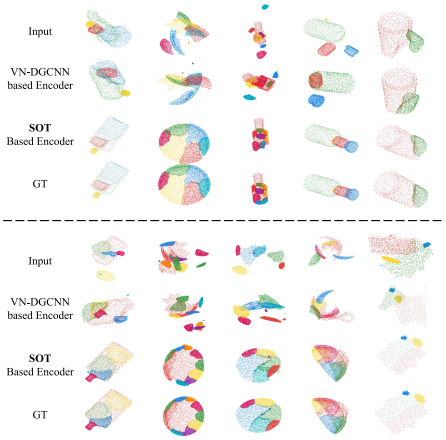
Encoder	Artifact					
	RMSE-R ↓ (°)	RMSE-T ↓ (10^{-2})	CD ↓ (10^{-2})	CRD ↓ (10^{-2})	PA_{CD} ↑ (%)	PA_{CRD} ↑ (%)
VN-DGCNN	34.90	9.34	13.75	16.20	67.64	60.18
SOT (Ours)	**34.16**	**8.97**	**12.74**	**15.56**	**68.89**	**62.76**



(a) SE(3)-Assembly across different encoders.

(b) Multi-part shape assembly results.

Encoder	Breaking Bad (Random Translation)			
	RMSE-R ↓	RMSE-T ↓	CD ↓	PA ↑
VN-DGCNN	69.3440	0.1138	0.2001	0.7350
SOT (Ours)	**43.3528**	**0.0768**	**0.1435**	**0.7984**



(b) CMNet across different encoders.

Fig. 4: Qualitative results for (a) 2BY2 dataset using SE(3)-Assembly, and (b) Breaking Bad dataset with CMNet.

Baselines and Encoders. To demonstrate that our encoder is broadly applicable, we evaluate it with pose regression-based works such as NSM [7], SE(3)-Assembly [42], and 2BY2NET [28]. For baselines, we consider three widely used encoders: DGCNN [37], VN-DGCNN [10], and the dual-branch feature extractor branch from FINet [44] (FINet-EB). Unless otherwise noted, we replace each model's original encoder with ours while keeping all other components.

Implementation Details. Unless otherwise specified, we train for 100 epochs on 2BY2 [28] and Breaking Bad [32], and for 200 epochs on ShapeNet [6]. NSM [7], SE(3)-Assembly [42], and 2BY2NET [28] use Adam (learning rate 1×10^{-4}, weight decay 1×10^{-6}) with no learning-rate scheduler. Jigsaw [27] uses SGD (learning rate 1×10^{-3}, no weight decay) with a cosine schedule that decays the learning rate from 1×10^{-3} to 1×10^{-5}. All two-part assembly experiments run on four NVIDIA GeForce RTX 3090 GPUs with a total batch size of 64. Also, to train CMNet, we follow the experimental protocol described in the [18] for the artifact category of the Breaking Bad dataset, except that we train for 90 epochs rather than 300 on four NVIDIA GeForce RTX 3090 GPUs with a total batch size of 32. Furthermore, Multi-part assembly experiment [42] uses 4 × NVIDIA GeForce RTX 3090 GPUs with a total batch size of 32 with SE(3)-Assembly on the everyday category.

Models	Enc.	RMSE-R ↓	RMSE-T ↓	CD ↓	PA ↑	$\mathcal{D}_{rt-r}$ ↓
SE(3)-Assembly	Ours (w/o $\mathcal{L}_w$)	34.0900	0.3161	0.9088	0.5687	0.0682
	Ours (w/o DS)	32.8855	0.3315	0.7656	0.5376	0.0427
	Ours (w/o Q_t)	32.8937	0.3352	0.7042	0.5652	-
	Ours	**32.3640**	**0.3039**	**0.6200**	**0.5911**	**0.0381**

Table 3: Ablation study. Shape assembly task on 2BY2 dataset. DS indicates direct subtraction.

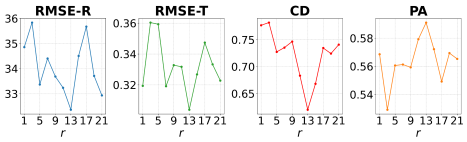


Fig. 5: Hyper-parameter tuning for r.

Table 4: We conduct a comparative study by instantiating the rotation branch on 2BY2 dataset with VN, VNT, and RETI-VN, respectively.

Encoder	f'	SE(3)-Assembly RMSE-R ↓	SE(3)-Assembly RMSE-T ↓	SE(3)-Assembly CD ↓	SE(3)-Assembly PA ↑	NSM RMSE-R ↓	NSM RMSE-T ↓	NSM CD ↓	NSM PA ↑
SOT	VNT	84.4165	0.3414	4.1046	0.5216	88.3118	0.5038	5.4037	0.2974
	VN	38.0648	0.3370	1.2781	0.5461	51.1724	0.5476	0.9560	0.5047
	RETI-VN	**32.3640**	**0.3039**	**0.6200**	**0.5911**	**29.1815**	**0.3982**	**0.3490**	**0.6121**

5.2 Shape Assembly

Two-Part Quantitative Results. Our encoder consistently outperforms those used in competitive method, including DGCNN [37], VN-DGCNN [10], and FINet-EB [44], as shown in Tab. 1. The improvements are most pronounced on the large-scale 2BY2 and Breaking Bad datasets. When integrated into pose regression pipelines [7,28,42], the performance gains become even larger. In particular, we observe substantial improvements in CD and RMSE-T, indicating more accurate translation estimation.

In addition, we apply our method to CMNet [18], a dense feature matching approach, and train it on the artifact category. We replace the VN-DGCNN-based encoder of CMNet with an SOT-based encoder, and follow the original CMNet evaluation protocol, including the same set of reported metrics. As shown in Tab. 2a, our method leads to consistent improvements across all metrics on the Breaking Bad dataset.

Multi-Part Quantitative Results. We further evaluate our encoder on the multi-part assembly setting, where each object is composed of 2–20 parts. As shown in Tab. 2b, integrating our encoder into the SE(3)-Assembly [42] leads to consistent improvements across all metrics on the everyday category of the Breaking Bad dataset.

Shape Assembly Qualitative Results. As shown in Fig. 4, our model performs well across both direct pose regression [42] and dense feature matching [18] methods. Moreover, these results demonstrate that SOT reliably assembles diverse objects, even under varying translations in two-part shape assembly tasks.

5.3 Analysis on Architectural Components

Ablation Study. As shown in Tab. 3, removing $\mathcal{L}_w$ degrades all evaluation metrics, suggesting that the proposed weight regularization is beneficial for shape assembly. The comparison between Ours and Ours (w/o Q_t) suggests that making the translation suppression structure explicit via the projector $\mathbf{Q}_t\mathbf{Q}_t^\top$ is beneficial. Without Q_t, translation suppression is still encouraged by $\mathcal{L}_w$ but

Model	Param. ↓	RMSE-T ↓	RMSE-R ↓
PointNet	150.34K	0.0491	8.9837
DGCNN	309.63K	0.0550	7.9962
VN-DGCNN	72.21K	0.0321	8.9603
FINet-EB	77.86K	0.0313	8.9603
Ours	**34.79K**	**0.0304**	**6.1905**

Table 5: Pose regression on ModelNet40 with comparable model capacity.

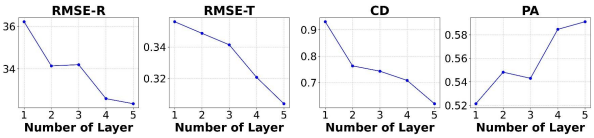


Fig. 6: Impact of stacking multiple RETI-VN layers.

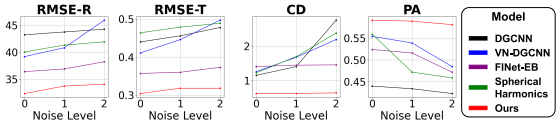


Fig. 7: Validation of noise robustness on the 2BY2 dataset. Noise levels 0, 1, and 2 correspond to Gaussian noise std. of 0.0, 0.01, and 0.025, respectively.

becomes less effective in practice. Finally, the gap between Ours and Ours (w/o DS) indicates that the direct subtraction step is beneficial in practice. This observation is consistent with the view that direct subtraction reduces the burden of directly learning $h(f(\mathbf{P})) = \mathbf{0}$.

Does $\mathcal{L}_w$ Suppress Translation? To further quantify translation leakage of RETI-VN f', we measure $\mathcal{D}_{rt-r} = ||f'(\mathbf{PR} + \mathbf{1}_n^\top \mathbf{t}) - f'(\mathbf{PR})||_2$. A larger value indicates higher sensitivity of f' to translation. As shown in Tab. 3, removing $\mathcal{L}_w$ increases $\mathcal{D}_{rt-r}$, indicating that $\mathcal{L}_w$ encourages reducing translation leakage by explicitly regularizing translation responses toward the translation suppression behavior in Eq. 2.

Impact of the Subspace Dimension of $\mathbf{Q}_t$. As noted in Sec. 4.1, the subspace dimension r of $\mathbf{Q}_t$ is treated as a hyperparameter. We therefore evaluate the model over a range of r values on the 2BY2 dataset. Fig. 5 shows that $r = 13$ provides the best trade-off across all metrics, yielding the lowest rotation error, translation error, and Chamfer distance, together with the highest part accuracy. This observation indicates that the most effective choice occurs near the middle of the output dimension size ($m = 21$).

Comparison among Vector Neurons. As shown by VNT [16], Eq. 1 removes translation and yields a rotation representation. Our RETI-VN layer is designed to achieve the same goal. To compare how well each layer captures rotational structure and generalizes, we construct f' as either VNT or RETI-VN and evaluate on the 2BY2 dataset. As reported in Tab. 4, RETI-VN consistently outperforms VNT. This indicates that RETI-VN extracts a rotation feature effectively for 3D shape assembly compared to VNT. It also suggests that the stronger weight constraint ($\sum_l \mathbf{W}_{kl} = 1$) in VNT may limit generalization [5].

Both RETI-VN and VN implement $SO(3)$-equivariant functions and produce rotation equivariant features. The RETI-VN consistently outperforms VN as shown in Tab. 4. This suggests that RETI-VN is the more effective $SO(3)$-equivariant layer for this task, as RETI-VN suppresses translation components under translation variations.

Pose Regression under Comparable Model Capacity. We additionally evaluate pose regression on the ModelNet40 dataset under comparable parameter budgets across baseline encoders (PointNet, DGCNN, VN-DGCNN, and FINet-EB) with the DCP model [36]. As shown in Tab. 5, our method attains the lowest translation and rotation error among all methods. Importantly, this result is achieved with only 34.79K parameters, indicating that our model achieves the most favorable balance of accuracy and parameter efficiency in this comparison.

Impact of RETI-VN depth. To investigate the architectural design of the RETI function f', we perform an ablation study on the number of RETI-VN layers used in the encoder. We implement f' as a composite mapping of RETI-VN and standard VN layers, and evaluate model performance while varying the number of RETI-VN layers on the 2BY2 dataset. For a given number k of RETI-VN layers, we set the first k encoder layers as RETI-VN and the remaining layers as standard VN layers. As shown in Fig. 6, increasing the number of RETI-VN layers consistently improves generalization, indicating that deeper RETI stacks more effectively suppress translation while preserving rotational structure.

Robustness to Noise. We evaluate the robustness to Gaussian noise (std.$\in \{0.0, 0.01, 0.025\}$) with SE(3)-Assembly [42], compared to DGCNN, VN-DGCNN, FINet-EB, and spherical harmonics based encoder, and ours on the 2BY2 dataset. As shown in Fig. 7, SOT is robust to noise compared to baselines.

5.4 Analysis of Disentangled Features

Experimental Setup. The SOT encoder yields two complementary representations: (i) the rotation and translation branches encourage $SO(3)$- and $T(3)$-equivariant features, respectively, and (ii) each branch is designed to be approximately invariant to the opposing factor. To evaluate these properties, we proceed as follows:

1. Apply a set of 3D rotations $\mathcal{R} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_n\}$ to a sample $\mathbf{P}_i$.
2. Extract rotation features from the rotation branch, $\mathcal{F}_i^r = \big\{f'(\mathbf{P}_i\mathbf{R}_1), f'(\mathbf{P}_i\mathbf{R}_2), \ldots, f'(\mathbf{P}_i\mathbf{R}_n)\big\}$.
3. Apply a set of 3D translations $\mathcal{TR} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$ to sample $\mathbf{P}_i$.
4. Extract translation features from the translation branch, $\mathcal{F}_i^t = \big\{\hat{f}(\mathbf{P}_i + \mathbf{1}_m^\top\mathbf{t}_1), \hat{f}(\mathbf{P}_i + \mathbf{1}_m^\top\mathbf{t}_2), \ldots, \hat{f}(\mathbf{P}_i + \mathbf{1}_m^\top\mathbf{t}_n)\big\}$, where $\hat{f}(\cdot) = f(\cdot) - f'(\cdot)$
5. Apply all steps to all samples $\mathcal{P} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N\}$.

Equivariance to the Target Component. We evaluate the quality of rotation- and translation-equivariance of the disentangled features. Because f' is implemented with RETI-VN layers, it satisfies $f'(\mathbf{P}_i\mathbf{R}) = f'(\mathbf{P}_i)\mathbf{R}$. For each sample $\mathbf{P}_i$ and rotation $\mathbf{R}_j$, we estimate the induced rotation in feature space as $\hat{\mathbf{R}}_j^i = \big(f'(\mathbf{P}_i)\big)^\dagger f'(\mathbf{P}_i\mathbf{R}_j)$, where $(\cdot)^\dagger$ denotes a right inverse (e.g., pseudo inverse) on the rotation subspace. We then measure the rotation equivariance error (REE) as follows: $\text{REE} = \frac{1}{Nn}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{n}(\mathbf{R} - \hat{\mathbf{R}}_j^i)^2}$, where $\text{REE} \in \mathbb{R}^{3\times 3}$.

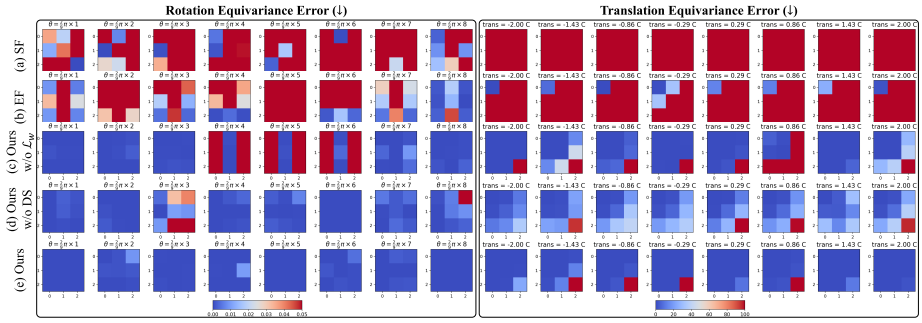


Fig. 8: Equivariance error under rotations and translations with SE(3)-Assembly for 2BY2 dataset. Left: rotation equivariance error for eight rotations. Right: translation equivariance error for translations scaled by the samples' centroid C. Rows: (a) SF (single feature) baseline, (b) EF (entangled feature) baseline, and (e) Ours. Here, $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$ denotes the rotation angles about the x, y, and z axes, and $\texttt{trans} \in \mathbb{R}^{1\times 3}$ is the translation vector. Darker blue indicates lower error. We extract an SF with a DGCNN and an EF with a FINet-EB.

Analogously, for translations $\{\mathbf{t}_j\}_{j=1}^n$, we define $\hat{\mathbf{t}}_j^{\,i} = \hat{f}\big(\mathbf{P}_i + \mathbf{1}_{n_i}^\top \mathbf{t}_j\big) - \hat{f}(\mathbf{P}_i)$, $\hat{f}(\cdot) = f(\cdot) - f'(\cdot)$, and compute the translation equivariance error (TEE) as follows: $\text{TEE} = \frac{1}{Nn}\sum_{i=1}^N \sqrt{\sum_{j=1}^n (\mathbf{t} - \hat{\mathbf{t}}_j^{\,i})^\top (\mathbf{t} - \hat{\mathbf{t}}_j^{\,i})}$, where $\text{TEE} \in \mathbb{R}^{3\times 3}$.

The left and right panels of Fig. 8 report REE and TEE, respectively. For perfectly rotation- or translation-equivariant features, the ideal scores are zero. In our experiments, the rotation and translation branches achieve values closest to zero, indicating that the proposed SOT encoder more faithfully recovers factor-wise equivariant representations for each component of the 3D pose than competing encoders. The proposed weight regularizer $\mathcal{L}_w$ and direct subtraction further improve the quality of equivariant representations, suggesting that our method disentangles rotation and translation more effectively than the baselines.

Invariance to the Counterpart Component. To verify that the SOT encoder of SE(3)-Assembly suppresses rotational content in the translation branch, we assess the quality of the translation features. We measure rotation-invariance by computing the average cosine similarity over all feature pairs in $\mathcal{F}_i^{t\prime} = \{\hat{f}(\mathbf{P}_i\mathbf{R}_1 + \mathbf{1}_m^\top \mathbf{t}_j), \hat{f}(\mathbf{P}_i\mathbf{R}_2 + \mathbf{1}_m^\top \mathbf{t}_j), \ldots, \hat{f}(\mathbf{P}_i\mathbf{R}_n + \mathbf{1}_m^\top \mathbf{t}_j)\}$, and then averaging this quantity over samples $\mathbf{P}_i \in \mathcal{P}$ to obtain R_{inv}: $\text{R}_{\text{inv}} = \frac{1}{Nn(n-1)}\sum_{i=1}^N \sum_{k,l\in\{0,\ldots,n\},\, k\neq l} \cos\Big(\hat{f}(\mathbf{P}_i\mathbf{R}_k + \mathbf{1}_m^\top \mathbf{t}_j),\ \hat{f}(\mathbf{P}_i\mathbf{R}_l + \mathbf{1}_m^\top \mathbf{t}_j)\Big)$. Analogously, we evaluate the translation-invariant features quality T_{inv} with $\mathcal{F}_i^{r\prime} = \big\{f'(\mathbf{P}_i + \mathbf{1}_m^\top \mathbf{t}_1),\ f'(\mathbf{P}_i + \mathbf{1}_m^\top \mathbf{t}_2),\ \ldots,\ f'(\mathbf{P}_i + \mathbf{1}_m^\top \mathbf{t}_n)\big\}$: $\text{T}_{\text{inv}} = \frac{1}{Nn(n-1)}\sum_{i=1}^N \sum_{k,l\in\{0,\ldots,n\},k\neq l} \cos(f'(\mathbf{P}_i + \mathbf{1}_m^\top \mathbf{t}_k), f'(\mathbf{P}_i + \mathbf{1}_m^\top \mathbf{t}_l))$. As shown in Tab. 6, SOT encoder with SE(3)-Assembly outperforms other encoders.

Table 6: Quality of disentangled features evaluation.

Encoder	2BY2	
	$\text{R}_{\text{inv}}\uparrow$	$\text{T}_{\text{inv}}\uparrow$
DGCNN	0.6614(±0.0916)	0.5852(±0.0282)
FINet-EB	0.2749(±0.1036)	0.5798(±0.0283)
SOT (w/o $\mathcal{L}_w$)	0.6979(±0.0418)	0.5932(±0.0416)
SOT (w/o DS)	0.7171(±0.0380)	0.5934(±0.0417)
SOT	**0.7707**(±0.0389)	**0.6132**(±0.0358)

Metric	R_{inv} ↑	T_{inv} ↑
SOT(1 RETI-VN)	0.6632(±0.0788)	0.5860(±0.0304)
SOT(2 RETI-VN)	0.6741(±0.0788)	0.5931(±0.0212)
SOT(3 RETI-VN)	0.7073(±0.0788)	0.6036(±0.0293)
SOT(4 RETI-VN)	0.7375(±0.0788)	0.6097(±0.0293)
SOT(5 RETI-VN)	**0.7707**(±0.0788)	**0.6132**(±0.0293)

Table 7: Quality of invariant features.

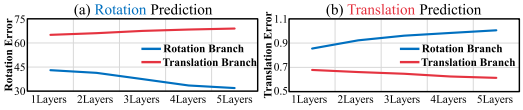


Fig. 9: Linear probing results across different numbers of layers.

This result implies that the rotation-branch features become less sensitive to translation changes, and the opposite holds for the translation branch.

Effect of Stacked RETI-VN Regularizer $\mathcal{L}_w$ softly suppresses translation responses, so a single RETI-VN layer may still retain residual translation leakage. As the number of RETI-VN layers increases, the invariance metric improves from 0.6632 to 0.7707 for R_{inv} and from 0.5860 to 0.6132 for T_{inv} (Tab. 7). Linear probing also shows that target-factor errors decrease, while non-target-factor errors increase across layers (Fig. 9). These results indicate that stacked RETI-VN progressively improves factor separation through repeated translation suppression.

6 Conclusion

We propose a simple yet effective encoder to encourage disentangling rotation and translation features within $SE(3)$-equivariant representations for 3D shape assembly. This encoder integrates seamlessly with a wide range of assembly pipelines and consistently improves performance across multiple datasets. Analyses further show that the learned features satisfy the desired equivariance and invariance properties, becoming more aligned with their target motion while being less sensitive to the other. These results suggest that explicitly suppressing non-target motion components provides a simple and effective inductive bias for 3D shape assembly.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2022R1A2C2012054, Development of AI for Canonicalized Expression of Trained Hypotheses by Resolving Ambiguity in Various Relation Levels of Representation Learning, Contribution Rate: 45.0%), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01842, Artificial Intelligence Graduate School Program (GIST), Contribution Rate: 5.0%), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2026-25495084, Contribution rate: 50.0%).

References

1. Ao, S., Hu, Q., Yang, B., Markham, A., Guo, Y.: Spinnet: Learning a general surface descriptor for 3d point cloud registration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11753–11762 (2021)
2. Bai, J., Kong, S., Gomes, C.: Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model. In: International Joint Conference on Artificial Intelligence (2020). https://doi.org/10.24963/ijcai.2020/595
3. Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C.L.: D3feat: Joint learning of dense detection and description of 3d local features. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6359–6367 (2020)
4. Bengio, Y., Courville, A.C., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**, 1798–1828 (2012), https://api.semanticscholar.org/CorpusID:393948
5. Bhojanapalli, S., Srebro, N.: Spectrally-normalized margin bounds for neural networks (2018)
6. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q.X., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. In: arXiv.org (2015)
7. Chen, Y.C., Li, H., Turpin, D., Jacobson, A., Garg, A.: Neural shape mating: Self-supervised object assembly with adversarial shape priors. In: Computer Vision and Pattern Recognition (2022). https://doi.org/10.1109/CVPR52688.2022.01239
8. Cheng, J., Wu, M., Zhang, R., Zhan, G., Wu, C., Dong, H.: Score-pa: Score-based 3d part assembly. arXiv preprint arXiv:2309.04220 (2023)
9. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2514–2523 (2020)
10. Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., Guibas, L.: Vector neurons: A general framework for so(3)-equivariant networks. In: IEEE International Conference on Computer Vision (2021). https://doi.org/10.1109/ICCV48922.2021.01198
11. Fuchs, F., Worrall, D.E., Fischer, V., Welling, M.: Se(3)-transformers: 3d roto-translation equivariant attention networks. In: Neural Information Processing Systems (2020)
12. Gandhi, S., Atul, Mahajan, S., Sharma, V., Gupta, R., Mondal, A.K., Singla, P.: Learning disentangled representation in object-centric models for visual dynamics prediction via transformers. In: arXiv.org (2024). https://doi.org/10.48550/arXiv.2407.03216
13. Huang, J., Zhan, G., Fan, Q., Mo, K., Shao, L., Chen, B., Guibas, L., Dong, H.: Generative 3d part assembly via dynamic graph learning. In: Neural Information Processing Systems (2020)
14. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: Predator: Registration of 3d point clouds with low overlap. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4267–4276 (June 2021)
15. Jiang, H., Salzmann, M., Dang, Z., Xie, J., Yang, J.: Se(3) diffusion model-based point cloud registration for robust 6d object pose estimation. In: Neural Information Processing Systems (2023). https://doi.org/10.48550/arXiv.2310.17359

16. Katzir, O., Lischinski, D., Cohen-Or, D.: Shape-pose disentanglement using se(3)-equivariant vector neurons. In: European Conference on Computer Vision (2022). https://doi.org/10.48550/arXiv.2204.01159
17. Lee, N., Min, J., Lee, J., Kim, S., Lee, K., Park, J., Cho, M.: 3d geometric shape assembly via efficient point cloud matching. ArXiv **abs/2407.10542** (2024)
18. Lee, N., Min, J., Lee, J., Park, C., Cho, M.: Combinative matching for geometric shape assembly. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
19. Lee, Y., Hu, E., Yang, Z., Yin, A., Lim, J.J.: Ikea furniture assembly environment for long-horizon complex manipulation tasks. In: IEEE International Conference on Robotics and Automation (2019). https://doi.org/10.1109/ICRA48506.2021.9560986
20. Li, J., Niu, C., Xu, K.: Learning part generation and assembly for structure-aware shape synthesis. In: AAAI Conference on Artificial Intelligence (2019)
21. Li, S., Zhu, J., Xie, Y.: Dbdnet: Partial-to-partial point cloud registration with dual branches decoupling. In: Knowledge-Based Systems (2023). https://doi.org/10.1016/j.knosys.2024.111864
22. Li, Y., Mo, K., Duan, Y., Wang, H., Zhang, J., Shao, L., Matusik, W., Guibas, L.: Category-level multi-part multi-joint 3d shape assembly. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3281–3291 (2023)
23. Li, Y., Mo, K., Shao, L., Sung, M., Guibas, L.: Learning 3d part assembly from a single image. In: European Conference on Computer Vision (2020). https://doi.org/10.1007/978-3-030-58539-6_40
24. Liu, H.Y., Guo, J.W., Jiang, H.Y., Liu, Y.C., Zhang, X.P., Yan, D.M.: Puzzlenet: Boundary-aware feature matching for non-overlapping 3d point clouds assembly. Journal of Computer Science and Technology **38**(3), 492–509 (2023)
25. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. In: Neural Information Processing Systems (2020)
26. Lu, J., Hua, G., Huang, Q.: Jigsaw++: Imagining complete shape priors for object reassembly. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6704–6714 (October 2025)
27. Lu, J., Sun, Y., Huang, Q.X.: Jigsaw: Learning to assemble multiple fractured objects. In: Neural Information Processing Systems (2023). https://doi.org/10.48550/arXiv.2305.17975
28. Qi, Y., Ju, Y., Wei, T., Chu, C., Wong, L.L., Xu, H.: Two by two: Learning multi-task pairwise objects assembly for generalizable robot manipulation (2025)
29. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K.: Geometric transformer for fast and robust point cloud registration. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11133–11142 (2022), https://api.semanticscholar.org/CorpusID:246823419
30. Scarpellini, G., Fiorini, S., Giuliari, F., Morerio, P., Bue, A.D.: Diffassemble: A unified graph-diffusion model for 2d and 3d reassembly. In: Computer Vision and Pattern Recognition (2024). https://doi.org/10.1109/CVPR52733.2024.02654
31. Schor, N., Katzir, O., Zhang, H., Cohen-Or, D.: Componet: Learning to generate the unseen by part synthesis and composition. In: IEEE International Conference on Computer Vision (2018)
32. Sell'an, S., Chen, Y.C., Wu, Z., Garg, A., Jacobson, A.: Breaking bad: A dataset for geometric fracture and reassembly. In: Neural Information Processing Systems (2022). https://doi.org/10.48550/arXiv.2210.11463

33. Sun, T., Zhu, L., Huang, S., Song, S., Armeni, I.: Rectified point flow: Generic point cloud pose estimation. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
34. Wang, X., Chen, H., Tang, S., Wu, Z., Zhu, W.: Disentangled representation learning. In: IEEE Transactions on Pattern Analysis and Machine Intelligence (2022). https://doi.org/10.1109/TPAMI.2024.3420937
35. Wang, Y., Solomon, J.: Prnet: Self-supervised learning for partial-to-partial registration. arxiv 2019. arXiv preprint arXiv:1910.12240 (1910)
36. Wang, Y., Solomon, J.: Deep closest point: Learning representations for point cloud registration. In: IEEE International Conference on Computer Vision (2019). https://doi.org/10.1109/ICCV.2019.00362
37. Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., Solomon, J.: Dynamic graph cnn for learning on point clouds. In: ACM Transactions on Graphics (2018). https://doi.org/10.1145/3326362
38. Wang, Z., Chen, J., Furukawa, Y.: Puzzlefusion++: Auto-agglomerative 3d fracture assembly by denoise and verify. In: arXiv.org (2024). https://doi.org/10.48550/arXiv.2406.00259
39. Wang, Z., Jörnsten, R.: Se(3)-bi-equivariant transformers for point cloud assembly. In: Neural Information Processing Systems (2024). https://doi.org/10.48550/arXiv.2407.09167
40. Wang, Z., Xue, N., Jörnsten, R.: Equivariant flow matching for point cloud assembly. ArXiv **abs/2505.21539** (2025), https://api.semanticscholar.org/CorpusID:278959897
41. Willis, K.D.D., Jayaraman, P., Chu, H., Tian, Y., Li, Y., Grandi, D., Sanghi, A., Tran, L.T., Lambourne, J., Solar-Lezama, A., Matusik, W.: Joinable: Learning bottom-up assembly of parametric cad joints. In: Computer Vision and Pattern Recognition (2021). https://doi.org/10.1109/CVPR52688.2022.01539
42. Wu, R., Tie, C., Du, Y., Zhao, Y., Dong, H.: Leveraging se(3) equivariance for learning 3d geometric shape assembly. In: IEEE International Conference on Computer Vision (2023). https://doi.org/10.1109/ICCV51070.2023.01316
43. Wu, R., Zhuang, Y., Xu, K., Zhang, H., Chen, B.: Pq-net: A generative part seq2seq network for 3d shapes. In: Computer Vision and Pattern Recognition (2019)
44. Xu, H., Ye, N., Liu, S., Liu, G., Zeng, B.: Finet: Dual branches feature interaction for partial-to-partial point cloud registration. In: AAAI Conference on Artificial Intelligence (2021). https://doi.org/10.1609/aaai.v36i3.20189
45. Xu, J., Le, H.M., Huang, M., Athar, S., Samaras, D.: Variational feature disentangling for fine-grained few-shot classification. In: IEEE International Conference on Computer Vision (2020). https://doi.org/10.1109/ICCV48922.2021.00869
46. Yew, Z.J., Lee, G.H.: Rpm-net: Robust point matching using learned features. In: Computer Vision and Pattern Recognition (2020). https://doi.org/10.1109/cvpr42600.2020.01184
47. Yu, H., Li, F., Saleh, M., Busam, B., Ilic, S.: Cofinet: Reliable coarse-to-fine correspondences for robust point cloud registration. In: Neural Information Processing Systems (2021)