

Unsupervised Pixel-Level Semantic Left-Right Understanding of In-the-Wild Images

Weikang Wang^{1,2}, Tobias Weißberg^{1,2}, and Florian Bernard^{1,2}

¹ University of Bonn, Germany

² Lamarr Institute, Germany

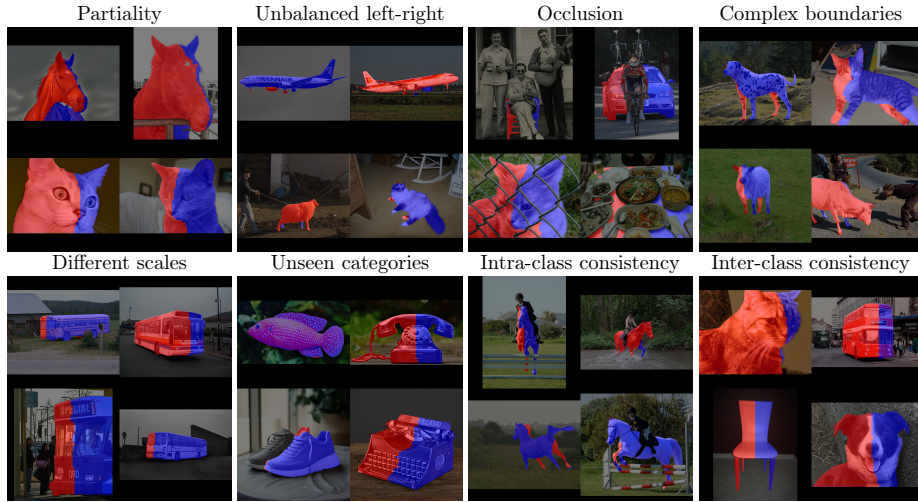


Fig. 1: We propose the first unsupervised *pixel-level* semantic left-right prediction framework for in-the-wild images, which is robust across diverse challenging settings.

Abstract. While various works address reflective symmetry understanding in 3D data and images, pixel-level semantic left-right prediction of in-the-wild images remains challenging, due to certain difficulties including the lack of 3D information, occlusion, object pose variation, partiality, *etc.* In this work, we propose an unsupervised learning framework to tackle this challenge. Leveraging recent advances in vertex-wise semantic left-right understanding of 3D data, our unsupervised learning method jointly utilises 3D shape and image datasets to infer pixel-wise semantic left-right predictions in single-view images. In particular, we show that a medium-scale 3D shape dataset comprising mainly of human- and quadruped animal-like shapes, combined with diverse in-the-wild image data, are sufficient to achieve high-quality semantic left-right prediction in images, even for entirely unseen 3D object categories, such as cars or trains. Overall, our approach achieves superior performance in dense pixel-wise semantic left-right predictions on both rendered and in-the-wild image datasets when compared to existing state-of-the-art methods.

Keywords: Left-Right Symmetry · Semantic Image Understanding

1 Introduction

Reflective symmetry understanding, especially left-right symmetry understanding, has been a long standing topic that is studied in different areas of visual computing. The high relevance of left-right symmetry stems from the fact that it is a ubiquitous property observed in various object categories, including humans, animals and man-made objects such as cars, bicycles, or aeroplanes.

From a *geometric* perspective, reflective symmetries (including left-right symmetry) are conventionally classified as extrinsic or intrinsic ones [15, 19]. For 3D data, such as point clouds and meshes, numerous methods capable of detecting extrinsic [1, 9, 13, 31] or intrinsic reflective symmetries [10, 14, 19, 22, 23, 30] have demonstrated high-quality results.

In contrast, left-right understanding in 2D images is significantly more challenging. Left-right symmetry observed in images arises from the underlying 3D structure of the object they depict, and common difficulties including the lack of 3D information in single-view images, occlusion, pose variation and partiality, make geometric modelling of symmetries in 2D images ill-posed in many settings.

Existing methods attempt to address this challenge, but they possess certain limitations. One category of methods focuses on general symmetry axis detection [26, 27, 33, 34], yet they are restricted to objects exhibiting (near) perfect extrinsic symmetry and struggle to handle cases like occlusion or partiality. Another line of works [8, 29, 35] utilises semantic correspondence as a proxy to refine left-right aware image features; however, keypoints are required as supervision signals.

Recent works [30, 32] introduce a new *semantic* perspective to tackle left-right understanding problem. Emerging studies [3, 6, 16, 35] show vision foundation models encode rich left-right semantics. Leveraging this, [30, 32] formalise left-right understanding of 3D shapes as a vertex-wise semantic prediction task. By utilising shape descriptors decorated with features from vision foundation models [4], these methods extract reliable predictions without requiring annotated data, effectively introducing a semantic paradigm to left-right understanding.

Inspired by this progress in vertex-wise left-right understanding of 3D shapes, we propose Pix2LR, a flexible and straightforward unsupervised framework for pixel-wise semantic left-right prediction, to **explicitly detect the semantic left and right parts** of objects within images. We summarise our main contributions as follows:

- We propose the first unsupervised framework to predict dense pixel-level semantic left-right labels across a broad range of objects in in-the-wild images.
- To achieve this, we leverage a hybrid training that uses a medium-scale 3D shape dataset of limited object categories (human- and animal-like shapes), together with image datasets of a broader class of object categories.
- We design a per-vertex left-right prediction aggregation strategy to ensure the consistency for left-right predictions of shape and image data.
- We show the effectiveness of our dense pixel-level predictions for semantic left-right understanding, both on rendered and in-the-wild image datasets.
- Evaluation on unseen categories and image styles show the strong generalisation ability of our proposed framework.

2 Related Works

We briefly review works that are closely related to reflective symmetry and left-right understanding.

Left-Right understanding in 3D domain. Traditional symmetry detection methods that geometrically model reflective symmetries as invariant to reflection (extrinsic symmetries) or isometries (intrinsic symmetries) can be directly leveraged for left-right understanding of 3D data. In the most straightforward setting, extrinsic symmetry detection methods have already achieved high-quality results. Mitra *et al.* [18] propose an algorithm that matches simple local shape signatures in pairs and uses these matches to accumulate evidence for symmetries in an appropriate transformation space, followed by clustering and verification steps. Je *et al.* [9] improve upon [18] by utilizing langevin dynamics within an altered symmetry space to eliminate the verification step. E3Sym [12] achieves symmetry detection by extracting $E(3)$ -invariant features through a lightweight neural network, facilitating robust symmetry predictions.

As a more challenging task, intrinsic symmetry detection has garnered attention in recent decades. Ovsjanikov *et al.* [22] model intrinsic shape symmetries as extrinsic symmetries in eigenfunction signature space and detect symmetries by searching over all possible candidates. Nagar *et al.* [20] improve this algorithm by sampling sparse intrinsic symmetric vertex pairs and leveraging the fact that the shortest geodesic path between two intrinsically symmetric points is also intrinsically symmetric. Liu *et al.* [14] extract intrinsic symmetries on genus-zero meshes by extracting closed curves from conformal maps of extremal points.

Recently, methods specifically designed for semantic left-right understanding of 3D data have been proposed. Using shape vertex descriptors enriched with semantic features from vision foundation models [4], χ [30] detects per-vertex left-right symmetry values across different shape categories using a combination of geometric losses. Weißberg *et al.* [32] extend [30] by disentangling input vertex-based features into left-right agnostic and informative components, and improve the robustness of detected left-right symmetries.

Left-Right understanding in 2D domain. Due to the absence of 3D information in single-view 2D images, left-right understanding in 2D images is significantly more challenging compared to its 3D counterpart. While existing methods attempt to address this challenge, they suffer from various notable limitations. Several works [26, 27, 34] target at general symmetry axis detection. Seo *et al.* [27] detect symmetries by employing polar pooling, self-similarity encoding, and angle-specific kernels, combined with a self-supervised data augmentation technique. Equisym [26] uses an end-to-end framework for symmetry detection that leverages dihedrally-equivariant feature maps. Yu *et al.* [34] introduce a novel framework for axis-level extrinsic symmetry detection by representing symmetry axes as explicit geometric primitives, namely lines. While these symmetry axis prediction methods detect reasonable extrinsic symmetries and can be applied to left-right symmetry understanding, they exhibit poor performance on objects with occlusions, partialities, complex poses, and other challenging real-world conditions.

Another line of works [8, 29, 35] in the semantic correspondence community aim to fine-tune features from vision foundation models to be more left-right aware, typically by using annotated sparse keypoint pairs as supervisory signals. Although the refined features perform well in semantic correspondence, they are not capable of performing pixel-wise left-right understanding effectively.

Left-Right information from vision foundation models. Vision foundation models, such as DINOv2 [21], DINOv3 [28], CLIP [24], and Stable Diffusion [25], have demonstrated remarkable effectiveness across various image-based tasks. As a ubiquitous property spanning different object categories, left-right information embedded within vision foundation models has been analysed in recent research. El Banani *et al.* [6] demonstrate that directly utilising features from vision foundation models for certain tasks (*e.g.*, semantic correspondence) can sometimes lead to incorrect matching of similar structures (including left-right symmetric parts) while succeeding in other cases. This validates the existence of noisy left-right information within these features. Cheng *et al.* [3] optimise a spectral domain mapping derived from features of one vision foundation model to achieve consensus with features from another vision foundation model. Their results of correct left-right keypoint matching between images provide evidence for the existence of left-right information within vision foundation models. Zhang *et al.* [35] quantitatively validate the performance of different vision foundation models [21, 25] for object left-right pose estimation by computing similarities between features of input images and manually designed pose templates.

As previously discussed, two recent works on 3D shape semantic left-right understanding [30, 32] have successfully extracted left-right information from per-vertex features enriched with vision foundation models. In this work, operating under the similar assumption that vision foundation models contain rich left-right information, we employ two widely-used vision foundation models: Stable Diffusion [25] (abbreviated as SD) and DINOv3 [28].

To our knowledge, no existing method directly addresses pixel-wise left-right understanding of images semantically. Given the evidence that vision foundation models contain rich left-right semantics [3, 6, 35], and inspired by semantic left-right understanding of 3D data [30, 32], this work aims to bridge this gap.

3 Method

We first briefly recap the main ideas of χ [30], a recent state-of-the-art unsupervised method for vertex-wise semantic left-right prediction of intrinsically symmetric 3D shapes. Then, we elaborate (i) the hybrid training data preparation of unannotated 3D shape and in-the-wild image datasets, (ii) the dense pixel-wise left-right predictor, (iii) the per-vertex left-right prediction aggregation process, and (iv) a combination of unsupervised losses, of our pixel-wise semantic left-right prediction framework, Pix2LR. Our notation is summarised in Tab. 1.

Implementation details, including detailed network structure and training details, are provided in the supplementary material.

Table 1: Summary of the notation used in this paper.

Symbol	Description	Symbol	Description
$\mathcal{M} = (V, E)$	Triangle mesh	$F_r \in \mathbb{R}^{N_r \times H \times W \times D}$	Features of I_r
$V_{\mathcal{M}}$	Vertices of $\mathcal{M}$	$\bar{F}_r \in \mathbb{R}^{N_r \times H \times W \times D}$	Flipped features of $\bar{I}_r$
$E_{\mathcal{M}}$	Edges of $\mathcal{M}$	$F_w \in \mathbb{R}^{N_w \times H \times W \times D}$	Features of I_w
$I_r \in \mathbb{R}^{N_r \times H \times W \times 3}$	Rendered images	$\bar{F}_w \in \mathbb{R}^{N_w \times H \times W \times D}$	Flipped features of $\bar{I}_w$
$\bar{I}_r \in \mathbb{R}^{N_r \times H \times W \times 3}$	Flipped rendered images	$\mathcal{F}_{\mathcal{M}}, \bar{\mathcal{F}}_{\mathcal{M}} \in \mathbb{R}^{ V_{\mathcal{M}} \times D}$	Per-vertex features of $\mathcal{M}$
$I_w \in \mathbb{R}^{N_w \times H \times W \times 3}$	In-the-wild images	$S_w, \bar{S}_w \in [-1, 1]^{N_w \times H \times W}$	Per-pixel left-right values
$\bar{I}_w \in \mathbb{R}^{N_w \times H \times W \times 3}$	Flipped in-the-wild images	$\chi_{\mathcal{M}}, \bar{\chi}_{\mathcal{M}} \in [-1, 1]^{ V_{\mathcal{M}} }$	Per-vertex left-right values
$M_w \in \{0, 1\}^{N_w \times H \times W}$	Object masks of I_w	$\chi_v, \bar{\chi}_v \in [-1, 1]$	Left-right value of vertex v

3.1 Background

First, χ [30] follows Diff3F [4] to obtain vertex-wise shape features by aggregating image features from vision foundation model onto the 3D shape surface. Specifically, with an input triangle mesh $\mathcal{M} = (V_{\mathcal{M}}, E_{\mathcal{M}})$, N_r images are rendered from surrounding virtual cameras. And then the feature descriptor of each vertex is defined as the averaged image features of corresponding pixels across all N_r views (vertex-to-pixel correspondences are known due to the rendering process), forming per-vertex shape features $\mathcal{F}_{\mathcal{M}} \in \mathbb{R}^{|V_{\mathcal{M}}| \times D}$. Subsequently, χ [30] (i) flips each rendered image *horizontally* to create a virtual image (with swapped roles of left and right), (ii) extracts image features of those flipped images, (iii) reverts the horizontal flip of the feature images, and then aggregates the image-based features to get per-vertex feature descriptors $\bar{\mathcal{F}}_{\mathcal{M}} \in \mathbb{R}^{|V_{\mathcal{M}}| \times D}$. And finally, a lightweight network is utilised to learn a projection of the high-dimensional per-vertex features onto a one-dimensional chirality subspace (for both $\mathcal{F}_{\mathcal{M}}$ and $\bar{\mathcal{F}}_{\mathcal{M}}$), denoted as $\chi_{\mathcal{M}}, \bar{\chi}_{\mathcal{M}} \in [-1, 1]^{|V_{\mathcal{M}}|}$, respectively. By carefully designed losses that enforce left-right consistency between $\chi_{\mathcal{M}}$ and $\bar{\chi}_{\mathcal{M}}$ and certain geometrical constraints, it's demonstrated that no supervision is necessary to perform reliable per-vertex semantic left-right predictions. We summarise geometric losses that are relevant to this work in the the following. The *dissimilarity loss*

$$\mathcal{L}_{\text{dis}} = -\frac{1}{\sqrt{|V_{\mathcal{M}}|}} \|\chi_{\mathcal{M}} - \bar{\chi}_{\mathcal{M}}\|_2 \quad (1)$$

aims to enforce a large difference in the left-right predictions of intrinsically symmetric vertex pairs. The *total variation loss*

$$\mathcal{L}_{\text{var}} = \frac{1}{|E_{\mathcal{M}}|} \sum_{(u,v) \in E_{\mathcal{M}}} \|\chi_u - \chi_v\|_1 + \|\bar{\chi}_u - \bar{\chi}_v\|_1 \quad (2)$$

promotes spatial smoothness. And the *fifty-fifty loss*

$$\mathcal{L}_{\text{ff}} = \frac{1}{|V_{\mathcal{M}}|} \left(\frac{|\chi_{\mathcal{M}}^\top \mathbf{1}_{|V|}|}{\|\chi_{\mathcal{M}}\|_\infty} + \frac{|\bar{\chi}_{\mathcal{M}}^\top \mathbf{1}_{|V|}|}{\|\bar{\chi}_{\mathcal{M}}\|_\infty} \right) \quad (3)$$

favours predictions in which left and right parts have the same number of assigned vertices.

3.2 Pixel-wise Semantic Left-Right Prediction

Now we introduce our Pix2LR framework.

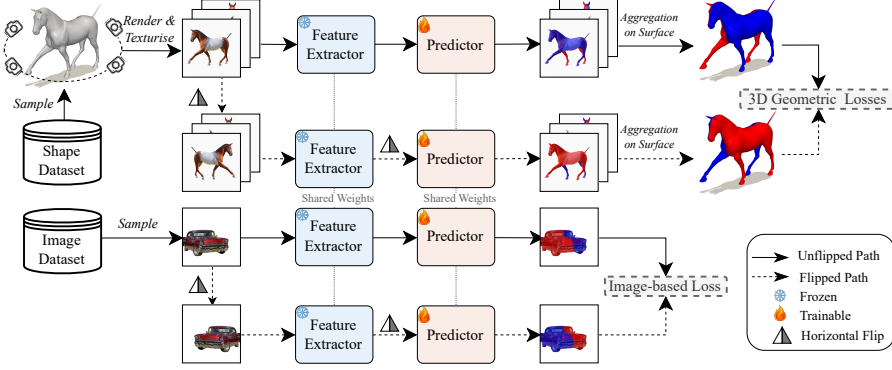


Fig. 2: Overview of our pixel-wise semantic left-right prediction framework Pix2LR. We use a hybrid training dataset comprising of a medium-scale 3D shape dataset (mainly of human- and animal-like shapes), which we use to render images, together with an in-the-wild image dataset (of a broad class of object categories, *e.g.*, including cars or aeroplanes). Rendered and in-the-wild images are both flipped horizontally to create images with roles of left and right swapped. Then we compute feature descriptors using a pretrained (and fixed) feature extractor for all images. These feature descriptors are then fed into the Predictor to predict left-right per pixel. To train Pix2LR, we combine 3D geometric losses defined on the 3D shape, together with an image-based loss that enforces symmetric consistency between the original and the flipped image.

Hybrid training data preparation. We utilise a medium-scale 3D shape dataset together with in-the-wild image datasets for our training. Based on [3, 6, 30, 32, 35], semantic left-right information contained in vision foundation models are common and consistent for semantically left-right symmetric objects from different categories. And we hypothesize that the missing or incomplete left-right 3D information within in-the-wild images can be supplied by 3D shapes, even from other categories. To this end, we carefully design a hybrid training strategy. Below we first elaborate the hybrid training data preparation process.

For each sampled 3D triangle mesh $\mathcal{M} = (V_{\mathcal{M}}, E_{\mathcal{M}})$ from the shape dataset, we render N_r images $I_r \in \mathbb{R}^{N_r \times H \times W \times 3}$ from surrounding views as in [4, 30, 32], and extract the features of those images using vision foundation models, denoted as $F_r \in \mathbb{R}^{N_r \times H \times W \times D}$.

Analogously, we use the same vision foundation models to extract features for N_w in-the-wild images $I_w \in \mathbb{R}^{N_w \times H \times W \times 3}$, sampled from the in-the-wild image dataset, denoted as $F_w \in \mathbb{R}^{N_w \times H \times W \times D}$. Here, we assume that object masks $M_w \in \{0, 1\}^{N_w \times H \times W}$ are available (*e.g.* obtained via some pre-trained mask detector such as SAM [11]) for those in-the-wild images I_w .

For any image either from rendered images set I_r or in-the-wild images set I_w , we conduct the following steps: (i) horizontally flip the original image, (ii) then use the same vision foundation models as above to extract features of the flipped image, (iii) and finally revert the horizontal flip of the extracted features of the flipped image. Fig. 3 (a) illustrates this process.

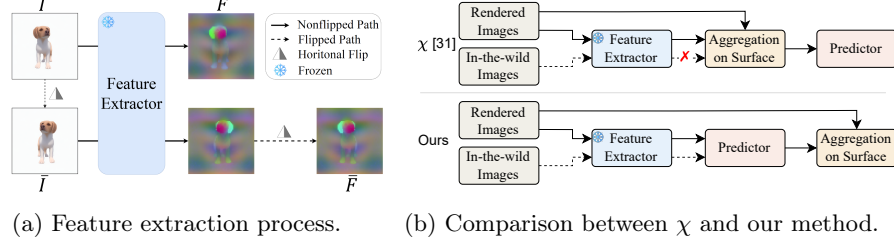


Fig. 3: (a) Given an image I , we use a shared vision foundation model to extract features from both I and its horizontally flipped counterpart $\bar{I}$. The feature obtained from $\bar{I}$ is then horizontally flipped back to align with I , yielding the final pair $(F, \bar{F})$. (b) Solid and dashed lines denote the prediction paths of rendered and in-the-wild images, respectively. The top branch illustrates χ [30] and the bottom branch shows our method. Since aggregation on the surface requires pixel-to-vertex correspondences, which are only available for rendered images, the path from in-the-wild images to predictor is unavailable in χ . Instead, we swap the order of left-right prediction and surface aggregation, so that our method keeps both paths unobstructed and produces consistent predictions for rendered and in-the-wild images.

After gathering features of the original images I_r, I_w , and flipped features of flipped images $\bar{I}_r, \bar{I}_w$, we obtain each batch of training data: $(F_r, \bar{F}_r, F_w, \bar{F}_w)$.

In the above process, we adopt horizontal image flipping rather than vertical or arbitrary axis flipping for two reasons: (1) Any 2D-axis flip is equivalent to a horizontal flip (plus an image in-plane rotation) and thus results in the same chirality change for the geometry of underlying 3D object the image depicts (*i.e.*, mirroring object in 3D space). (2) Our semantic features are extracted from vision foundation models whose training data overwhelmingly show upright objects (gravity bias). With that, these foundation models rarely see upside-down views (vertically flipped images), so the features of those views are less reliable. Previous works [4, 30, 32, 37] which leverage or extract features from vision foundation models, also incorporate this bias of camera positions. More explanations are in the supplementary materials.

Pixel-wise left-right predictor P_Φ . We adopt a light-weight network as our pixel-wise left-right predictor P_Φ , which takes image feature maps as input and outputs dense left-right predictions. Specifically, For the input features $F \in \mathbb{R}^{H \times W \times D}$, which can either be from the rendered images or in-the-wild images, the dense left-right predictions S is computed as $S = P_\Phi(F) \in [-1, 1]^{H' \times W'}$,

where $H'/H = W'/W$ represents possible scaling. Details about predictor structure are in the supplementary materials.

Per-vertex left-right prediction aggregation. We feed features of N_r rendered images, $F_r, \bar{F}_r$, into the predictor P_Φ to get pixel-wise left-right predictions $S_r, \bar{S}_r \in [-1, 1]^{N_r \times H' \times W'}$. Then, since vertex-to-pixel correspondence of each rendered image can be computed from virtual camera intrinsic and extrinsics, we decorate each vertex $v \in V_{\mathcal{M}}$ with the left-right value χ_v by averaging left-right predictions of its corresponding pixels from all N_r views, *i.e.*,

$$\chi_v = \frac{1}{\sum_{n=1}^{N_r} |\mathcal{I}_n^v|} \sum_{n=1}^{N_r} \sum_{(x,y) \in \mathcal{I}_n^v} S_r[n, x, y], \quad (4)$$

where $\mathcal{I}_n^v$ denote the set of corresponding pixels of rendered image $n \in 1, 2, \dots, N_r$ for vertex v . We stack the per-vertex left-right value $\chi_v \in [-1, 1]$ of all vertices to form $\chi_{\mathcal{M}} \in [-1, 1]^{|V_{\mathcal{M}}|}$. Similarly, $\bar{\chi}_{\mathcal{M}} \in [-1, 1]^{|V_{\mathcal{M}}|}$ is computed using $\bar{S}_r$.

Note that, similar to χ [30], we also obtain the per-vertex left-right values $\chi_{\mathcal{M}}, \bar{\chi}_{\mathcal{M}}$, but we do so in a different way by swapping the order of left-right prediction and aggregation. Right sub-figure in Fig. 3 (b) explains the difference in details. By incorporating this new design, we enable the framework to perform semantic left-right understanding of different data modes (images and 3D triangle meshes) consistently, thus make joint training of hybrid data be feasible.

Unsupervised losses. To train our model, we use a combination of 3D geometric losses (imposed on the 3D shapes) and an image-based loss (imposed on the in-the-wild images). Specifically, we consider

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{dis}} + \lambda_2 \mathcal{L}_{\text{var}} + \lambda_3 \mathcal{L}_{\text{fif}} + \lambda_4 \mathcal{L}_{\text{img}}, \quad (5)$$

where $\mathcal{L}_{\text{dis}}, \mathcal{L}_{\text{var}}$ and $\mathcal{L}_{\text{fif}}$ are as defined in Eq. (1), Eq. (2) and Eq. (3), computed using $\chi_{\mathcal{M}}$ and $\bar{\chi}_{\mathcal{M}}$ from above. $\mathcal{L}_{\text{img}}$ is the image-based loss:

$$\mathcal{L}_{\text{img}} = -\frac{1}{N_w} \|(S_w - \bar{S}_w) \cdot M_w\|_F, \quad (6)$$

where $S_w, \bar{S}_w \in [-1, 1]^{N_w \times H' \times W'}$ are the left-right predictions got from the predictor P_Φ with $F_w, \bar{F}_w$ as input.

4 Experiments

In this section we experimentally analyse on various image datasets the quality of pixel-wise left-right predictions from our proposed framework Pix2LR.

Shape datasets. We use the recently proposed medium-scale shape dataset BECoS [5]. BECoS [5] includes different human-like and quadruped animals shapes of various species with a train/test/val split of size 1975/284/274. It also provides consistent per-vertex ground truth left-right annotations that can be used for evaluation. In our ablation study, we also consider a popular human shape dataset FAUST [2], which contains 100 human shapes with train/test/val

split size of 80/10/10. Since FAUST [2] does not provide left-right annotations, we use respective annotations offered by BECOS [5] framework.

Image datasets. We use two common multi-category in-the-wild image datasets PF-PASCAL [7] and SPAIR-71K [17]. PF-PASCAL [7] contains images from 20 categories with 1001/506/521 train/test/val, with similar viewpoints and instance poses within each category. SPAIR-71K [17] contains images with large intra-class pose variation, with train/test/val as 997/481/322 and 18 categories.

In order to fully evaluate the performance of our proposed framework, we use both in-the-wild images from PF-PASCAL [7] and SPAIR-71K [17] test sets, as well as images rendered from BECOS [5] test shapes.

For rendered images of BECOS [5] test shapes, pixel-wise left-right annotations can be obtained from vertex-wise left-right annotations of shapes, as shown in the right sub-figure of Fig. 4. For in-the-wild images from either PF-PASCAL [7] or SPAIR-71K [17], left-right annotations are not available. By leveraging the consistent keypoint annotations within each category of these two datasets, we add new keypoint-wise left-right annotations to them. Specifically, for each category, we manually classify the keypoints into left side, right side and on the left-right boundary of each object (the left and right side is defined in the object view), and labelling them as 1, -1 and 0 respectively. Left sub-figure of Fig. 4 gives examples of these newly introduced keypoint-wise left-right annotations. More details about the annotation are in the supplementary materials.



Fig. 4: *Left:* Examples of *sparse* keypoint-wise left-right annotations of in-the-wild images from SPAIR-71K [17] and PF-PASCAL [7] datasets. Blue and red denote points within left and right parts, and white denotes the points on left-right boundary. *Right:* Examples of *dense* pixel-wise left-right annotations in rendered images from BECOS [5] dataset. Blue and Red denote semantic left and right parts.

Metrics. For either dense pixel-wise left-right annotations on rendered images from BECOS [5], or sparse keypoint-wise left-right annotations on in-the-wild images from PF-PASCAL [7] and SPAIR-71K [17], we use the same accuracy metric as defined below. Specifically, for experiments over N images:

$$acc = \max\left(\frac{1}{N} \sum_{i=1}^N \frac{\mathbb{1}(y_i \cdot y_i^{\text{gt}} > 0)}{\|y_i \cdot y_i^{\text{gt}}\|_2^2}, \frac{1}{N} \sum_{i=1}^N \frac{\mathbb{1}(y_i \cdot y_i^{\text{gt}} < 0)}{\|y_i \cdot y_i^{\text{gt}}\|_2^2}\right), \quad (7)$$

where $y_i \in \{-1, 1\}^{M_i}$ and $y_i^{\text{gt}} \in \{-1, 0, 1\}^{M_i}$ denote the predicted and ground-truth left-right values of all keypoints or all foreground pixels of image i , respec-

tively. Note that the size M_i depend on individual image i . The final accuracy acc is the maximum between two values above since the roles with left-right swapped are still valid as long as they are consistent within this category.

Baselines. We compare the performance of our framework with several state-of-the-art left-right aware feature descriptors: SD+DINOv2 [36] and its variant SD+DINOv3, GeoAware [35], as well as a recent vertex-wise left-right predictor of 3D shapes χ [30]. Since χ [30] is originally designed to predict per-vertex left-right values independently, we adopt a pretrained χ [30] on BECOS (using the same vision foundation models as ours for fair comparison), denoted as χ_{BeCoS} , to perform per-pixel left-right predictions with per-pixel features.

For each feature descriptor, different steps are conducted for in-the-wild images with sparse keypoint-wise left-right annotations, and for rendered images with dense pixel-wise left-right annotations, to obtain predictions. For in-the-wild images, the steps are as follows. For each category, (1). First, for each keypoint, compute its descriptor by averaging its corresponding features from all training images. Denote them as template keypoints. (2). Then, apply a 2-centre clustering of all template keypoints without the ones labelled as 0 (keypoints on the left-right boundary) using their descriptors, and label them with clustered labels as being -1 or 1 (keypoints on the left-right boundary labelled as 0). (3). Finally, for each keypoint within test images, assign it with the label of its closest template keypoint, in the sense of cosine similarity between its feature and template keypoint descriptors. For rendered images, the steps are as follows. As in [4, 30, 32], we decorate each shape vertex with the averaged image features of corresponding pixels from all N_r rendered images. Then for each pixel within a test image, assign it with the left-right label of its closest vertex, in the sense of cosine similarity between features.

4.1 Pixel-level Left-Right Prediction on Real Images

For images within each category from PF-PASCAL [7] or SPAIR-71K [17], we use the sparse keypoint-wise annotations defined as above, to compute the accuracy acc over all test images within this category using Eq. (7), with N being the number of images of this category, K_c being the number of keypoints of this category, and $y_i \in \{-1, 1\}^{K_c}$, $y_i^{\text{gt}} \in \{-1, 0, 1\}^{K_c}$ being the prediction and ground truth of all keypoints of image i , respectively.

We also compute an overall accuracy $\hat{acc}$ across all categories using Eq. (7), in order to evaluate the inter-class consistency, with N being the number of images of all categories, K_i being the number of keypoints of category for image i , and $y_i \in \{-1, 1\}^{K_i}$, $y_i^{\text{gt}} \in \{-1, 0, 1\}^{K_i}$ denoting the predictions and ground truth of image i , respectively.

Tab. 2 and Tab. 3 conclude the quantitative results of left-right prediction on SPAIR-71K [17] and PF-Pascal [7], respectively. For each dataset, we train our model using BeCoS [5] with it for fair comparison, denoted as $\text{Ours}_{\text{BeCoS-SPAIR-71K}}$ and $\text{Ours}_{\text{BeCoS-PF-Pascal}}$, respectively. We also train a model using BECOS [5] with both datasets, denoted as $\text{Ours}_{\text{BeCoS-Both}}$. We include qualitative comparison be-

Table 2: Left-Right predictions on SPAIR-71K [17]. Our method significantly outperforms its competitors for most object categories. For categories in which the competitive baseline GeoAware [35] performs better, our method still gives comparable numbers. And for other categories and on average, our method outperforms GeoAware [35] with a large margin. We observe that training ours on both image dataset only leads to slightly better results on average compared to using only SPAIR-71K [17] as image dataset.

Method																			AVG <i>acc</i>	<i>acc</i>
SD+DINO [36]	56.6	51.2	50.6	57.2	50.4	51.5	50.5	51.8	52.5	51.3	51.3	50.0	55.7	51.0	52.2	50.5	50.2	50.3	51.9	50.0
SD+DINOv3	55.2	51.2	50.6	57.2	50.9	50.8	50.8	51.4	52.9	51.8	50.1	50.0	56.7	51.0	96.9	50.5	50.5	96.1	57.2	55.8
GeoAware [35]	96.5	53.1	50.6	57.9	90.4	51.8	50.5	51.4	89.7	95.6	51.3	51.0	70.3	94.4	100	50.5	50.1	100	70.1	64.0
χ_{BeCoS} [30]	72.2	69.4	79.8	55.5	73.4	72.0	61.9	84.5	73.0	80.5	79.3	73.9	53.0	82.6	75.5	74.8	79.1	84.1	73.6	73.0
Ours _{BeCoS-SPAIR71K}	76.9	67.4	81.0	52.0	89.7	93.4	89.6	83.2	86.7	85.0	81.1	76.2	71.0	89.2	98.2	76.0	90.4	99.0	82.7	82.7
Ours _{BeCoS-Both}	83.9	68.1	84.8	51.5	89.2	93.6	91.3	83.9	86.9	86.4	80.3	80.8	73.0	93.1	92.5	76.7	92.3	98.2	83.8	83.8

Table 3: Left-Right predictions on PF-PASCAL [7]. Our method significantly outperforms its competitors for most object categories. We observe that training ours on the PF-PASCAL [7] image dataset only leads to slightly better results on average compared to using both SPAIR-71K [17] and PF-PASCAL [7] as image dataset.

Method																			AVG <i>acc</i>	<i>acc</i>		
SD+DINO [36]	55.4	63.0	60.4	50.0	50.0	50.3	53.5	52.0	51.3	50.0	50.8	50.0	50.9	63.8	50.1	50.0	50.0	53.5	52.5	51.5	51.9	
SD+DINOv3	55.4	61.0	60.4	50.0	56.3	50.3	53.5	52.0	52.7	50.0	50.0	51.8	50.9	70.2	50.1	50.0	100	100	54.0	100	59.3	56.8
GeoAware [35]	96.1	71.0	52.1	50.0	90.6	50.3	50.6	51.4	93.4	50.0	50.0	50.0	55.3	80.9	50.5	50.0	100	53.0	51.0	100	63.2	52.2
χ_{BeCoS} [30]	76.2	57.0	95.8	63.6	66.7	78.8	72.4	97.7	82.9	72.9	63.8	96.2	81.3	60.6	93.2	80.8	100	85.2	67.2	79.0	77.3	77.3
Ours _{BeCoS-PFPascal}	90.9	50.0	100	100	95.9	90.2	88.3	99.2	89.1	100	93.8	99.5	83.2	60.2	96.7	98.2	100	95.3	93.9	100	86.8	86.8
Ours _{BeCoS-Both}	90.0	50.0	93.7	100	94.6	90.0	88.5	98.9	89.1	100	94.8	99.5	82.4	59.2	96.7	98.2	100	93.8	95.3	100	86.3	86.3

tween the most competitive baseline χ_{BeCoS} [30] and Ours_{BeCoS-Both} in Fig. 5. More qualitative results are in supplementary materials.

4.2 Pixel-level Left-Right Prediction on Rendered Images

For rendered images from test shapes of BECOS [5], for each category, the accuracy *acc* is defined as in Eq. (7), with N being the number of images within this category, $y_i \in \{-1, 1\}^{H \times W}$ the pixel-wise prediction, and $y_i^{\text{gt}} \in \{-1, 0, 1\}^{H \times W}$ the ground truth of image i . For a pixel p outside the object mask, $y_i^{\text{gt}}[p] = 0$.

For evaluating inter-class consistency of the semantic left-right predictions, we also include an overall accuracy *acc* across all categories of rendered images from BECOS [5] based on Eq. (7), with N being the number of images of all categories.

Tab. 4 summarises the performance on left-right prediction of all three versions of our model (Ours_{BeCoS-SPAIR-71K}, Ours_{BeCoS-PF-Pascal} and Ours_{BeCoS-Both}) and baselines, for each category within BECOS [5] test dataset. Qualitative results are included in the supplementary materials.

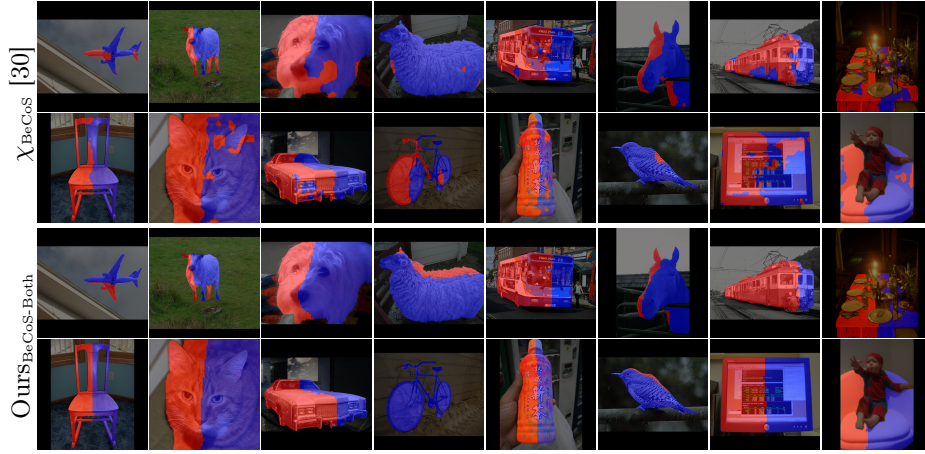


Fig. 5: Qualitative comparisons of dense pixel-level left-right predictions for χ_{BeCoS} [30] and $\text{Ours}_{\text{BeCoS-Both}}$ for different object categories sampled randomly from both SPAIR-71K [17] and PF-PASCAL [7]. Our predictions are significantly more accurate.

4.3 Generalisation Analysis

We analyse our trained framework in the following multiple levels to show its strong generalisation ability.

Generalisation on 3D information. The 3D shape dataset BECoS [5] that we use as 3D prior only contains human and quadruped animal shapes. As shown in Tab. 2, 3 and Fig. 5, our Pix2LR also performs well on images containing human-made objects such as aeroplanes, trains, and TVs, despite the fact that no explicit 3D information about these objects is used during training.

Generalisation on image geometry variations. As shown in Fig. 5, images within SPAIR-71K [17] and PF-PASCAL [7] contain images of objects

Table 4: Left-Right predictions on rendered images. Our method outperforms its competitors by a large margin. Since in this setting the images are rendered from the 3D shape dataset BECoS [5], the performance is similar independent of the choice of in-the-wild image training set.

Method										AVG <i>acc</i>	<i>a</i> <i>cc</i>
SD+DINO [36]	81.8	82.3	82.8	83.5	82.9	87.5	82.1	79.9	84.8	83.7	83.7
SD+DINOv3	78.1	79.8	78.7	79.9	79.6	84.1	79.0	74.8	80.5	80.4	80.4
GeoAware [35]	72.8	69.1	71.7	74.5	73.9	77.5	74.4	73.3	75.1	73.7	73.7
χ_{BeCoS} [30]	70.8	69.8	70.0	69.4	70.3	75.2	75.2	73.9	73.1	71.4	71.4
$\text{Ours}_{\text{BeCoS-SPair71k}}$	83.1	89.2	85.2	87.1	88.0	88.5	92.3	72.7	92.9	87.3	87.3
$\text{Ours}_{\text{BeCoS-PFPascal}}$	83.3	89.1	85.2	87.1	88.1	88.4	92.3	73.8	92.4	87.3	87.3
$\text{Ours}_{\text{BeCoS-Both}}$	83.7	89.9	84.9	87.3	88.2	87.9	92.4	75.8	93.7	87.4	87.4

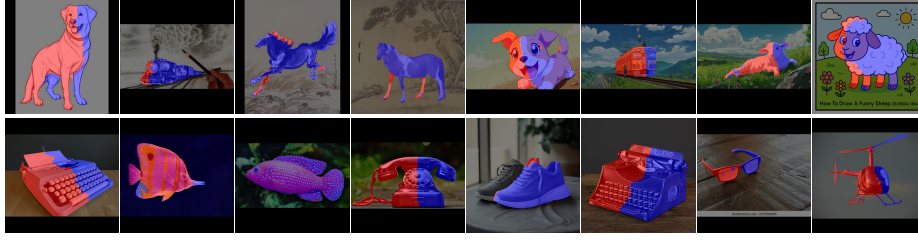


Fig. 6: Zero-shot performance of our method on paintings or anime images (*top*), and on unseen categories neither part of the 3D shape nor the image dataset (*bottom*).

from different categories with large pose and viewpoint variations, as well as occlusion and truncation. Yet, our framework can still performs well on most of these cases, which shows generalisation on images with large geometry variations.

Generalisation on image style variations. We further test the performance of our trained PixLR on images with different styles, such as anime or paintings. As shown in the first row of Fig. 6, our model can still predict reasonable left-right semantic labels, proves its generalisation ability on image styles.

Generalisation on unseen categories. We evaluate our trained Pix2LR on categories that are neither part of the 3D shape dataset, nor of the image dataset. Second row in Fig. 6 shows that the results are reasonable and consistent with the other object categories.

4.4 Ablation Analysis

We first conduct an ablation analysis on losses by leaving out individual loss terms. Further, we study the relevance of the employed 3D shape dataset used as 3D geometry prior during training. We use the left-right prediction accuracy $\hat{acc}$ of all categories as metrics for either in-the-wild images from SPAIR-71K [17] or rendered images from BECOS [5], for fair and comprehensive evaluation. Results are reported in Tab. 5.

Table 5: Ablation analysis when leaving out individual loss terms (*left*), and when training on different 3D shape datasets (*right*). The numbers represent $\hat{acc}$ on key-points and pixels, respectively. We confirm the necessity of each proposed loss, and conclude that adopting 3D shapes from more categories as in BECOS [5] indeed offers richer 3D priors. Note that although models trained without $\mathcal{L}_{img}$ achieves the same performance to the full model on rendered images, the large margin of performance on more challenging in-the-wild images from SPAIR-71K [17] validates its necessity.

	Ablation on losses				Ablation on 3D datasets	
	w/o $\mathcal{L}_{var}$	w/o $\mathcal{L}_{dis}$	w/o $\mathcal{L}_{lif}$	w/o $\mathcal{L}_{img}$	FAUST	BECoS
In-the-wild Images (SPAIR-71K [17])	55.6	56.8	60.9	73.8	78.2	83.8
Rendered Images (BECoS [5])	53.9	50.3	61.7	87.4	70.7	87.4



Fig. 7: We identify a few challenges for method leading to wrong predictions: (i) multiple symmetries, (ii) out-of-distribution views, (iii) unusual poses, (iv) rare geometries.

5 Discussion & Limitations

Overall, our Pix2LR shows superiority compared to other state-of-the-art methods for semantic left-right understanding of images across a broad range of categories. Nevertheless, there is room for improvements. First, as shown in Fig. 7, various hard cases may lead to inconsistent predictions. Future works that combine our framework with data augmentation, or leverage larger-scale 3D shape datasets with more versatile categories as geometric prior, may alleviate these difficulties. Besides, for some object categories which are rather ‘thin’ along the left-right axis, such as bicycle, it is hard to predict accurate left-right values of fine structures, such as handlebars or pedals, see Fig. 5. We believe that the addition of attention blocks for an instance-specific information exchange is an interesting direction. Finally, our current formalism only considers left-right symmetry, which is for example violated for rotationally symmetric objects or objects with more than two symmetries (*e.g.*, bottles or tables, in Fig. 5). Empirically, our methods produces consistent results in these cases – we observe that this ambiguity is implicitly resolved by interpreting the semantically ‘frontal part’ of the object to be facing the camera, so that visible pixels of respective objects are split more or less evenly into left and right regions. An interesting direction for future work is to consider more expressive symmetry representations.

6 Conclusion

The task of pixel-wise semantic left-right prediction of in-the-wild images is very challenging and under-explored. Difficulties include the absence of 3D information in single-view images, occlusion, as well as variations in object poses, viewpoints, texture, or geometry. In this work we propose the first unsupervised framework to explicitly detect left-right parts for objects within in-the-wild images from a broad set of categories. We tackle this by leveraging the recent medium-scale 3D shape dataset BECoS [5] to serve as a powerful geometric 3D prior, and then combine this with diverse in-the-wild images across different categories to train a left-right prediction neural network in an unsupervised manner. Our experiments qualitatively and quantitatively demonstrate the effectiveness of our proposed framework. We believe our work may be beneficial for multiple downstream applications, such as resolving left-right ambiguity in pose estimation, keypoint detection and part segmentation, or enforcing left-right consistency for image generation and editing.

Acknowledgements

This work is supported by the ERC starting grant no. 101160648 (Harmony). The authors gratefully acknowledge the access to the Marvin cluster of the University of Bonn.

References

1. Aguirre, I., Sipiran, I., Montanana, G.: A dataset-free approach for self-supervised learning of 3d reflectional symmetries. arXiv preprint arXiv:2503.02660 (2025)
2. Bogó, F., Romero, J., Loper, M., Black, M.J.: Faust: Dataset and evaluation for 3d mesh registration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3794–3801 (2014)
3. Cheng, X., Deng, C., Harley, A.W., Zhu, Y., Guibas, L.: Zero-shot image feature consensus with deep functional maps. In: European Conference on Computer Vision. pp. 277–293. Springer (2024)
4. Dutt, N.S., Muralikrishnan, S., Mitra, N.J.: Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4494–4504 (2024)
5. Ehm, V., Amrani, N.E., Xie, Y., Bastian, L., Gao, M., Wang, W., Sang, L., Cao, D., Weißberg, T., Löhner, Z., et al.: Beyond complete shapes: A benchmark for quantitative evaluation of 3d shape surface matching algorithms. In: Computer Graphics Forum. vol. 44, p. e70186. Wiley Online Library (2025)
6. El Banani, M., Raj, A., Maninis, K.K., Kar, A., Li, Y., Rubinstein, M., Sun, D., Guibas, L., Johnson, J., Jampani, V.: Probing the 3d awareness of visual foundation models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21795–21806 (2024)
7. Ham, B., Cho, M., Schmid, C., Ponce, J.: Proposal flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
8. Hartwig, R., Muhle, D., Marin, R., Cremers, D.: Geco: Geometrically consistent embedding with lightspeed inference. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9309–9319 (2025)
9. 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)
10. Kim, V.G., Lipman, Y., Chen, X., Funkhouser, T.: Möbius transformations for global intrinsic symmetry analysis. In: Computer graphics forum. vol. 29, pp. 1689–1700. Wiley Online Library (2010)
11. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
12. Li, R.W., Zhang, L.X., Li, C., Lai, Y.K., Gao, L.: E3sym: Leveraging e (3) invariance for unsupervised 3d planar reflective symmetry detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 14543–14553 (2023)
13. Li, X., Huang, Z., Thai, A., Rehg, J.M.: Symmetry strikes back: From single-image symmetry detection to 3d generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 743–752 (2025)

14. Liu, T., Kim, V.G., Funkhouser, T.: Finding surface correspondences using symmetry axis curves. In: *Computer Graphics Forum*. vol. 31, pp. 1607–1616. Wiley Online Library (2012)
15. Liu, Y., Hel-Or, H., Kaplan, C.S., Van Gool, L., et al.: Computational symmetry in computer vision and computer graphics. *Foundations and Trends® in Computer Graphics and Vision* **5**(1–2), 1–195 (2010)
16. Mariotti, O., Du, Z., Bhalgat, Y., Mac Aodha, O., Bilen, H.: Jamais vu: Exposing the generalization gap in supervised semantic correspondence. *Advances in Neural Information Processing Systems* **38**, 15841–15868 (2026)
17. Min, J., Lee, J., Ponce, J., Cho, M.: Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543* (2019)
18. Mitra, N.J., Guibas, L.J., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (ToG)* **25**(3), 560–568 (2006)
19. Mitra, N.J., Pauly, M., Wand, M., Ceylan, D.: Symmetry in 3d geometry: Extraction and applications. In: *Computer graphics forum*. vol. 32, pp. 1–23. Wiley Online Library (2013)
20. Nagar, R., Raman, S.: Fast and accurate intrinsic symmetry detection. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 417–434 (2018)
21. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research Journal* (2024)
22. Ovsjanikov, M., Sun, J., Guibas, L.: Global intrinsic symmetries of shapes. In: *Computer graphics forum*. vol. 27, pp. 1341–1348. Wiley Online Library (2008)
23. Qiao, Y.L., Gao, L., Liu, S.Z., Liu, L., Lai, Y.K., Chen, X.: Learning-based intrinsic reflectional symmetry detection. *IEEE Transactions on Visualization and Computer Graphics* **29**(9), 3799–3808 (2022)
24. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: *International conference on machine learning*. pp. 8748–8763. PmLR (2021)
25. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 10684–10695 (2022)
26. Seo, A., Kim, B., Kwak, S., Cho, M.: Reflection and rotation symmetry detection via equivariant learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 9539–9548 (2022)
27. Seo, A., Shim, W., Cho, M.: Learning to discover reflection symmetry via polar matching convolution. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 1285–1294 (2021)
28. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. *arXiv preprint arXiv:2508.10104* (2025)
29. Wandel, K., Wang, H.: Semalign3d: Semantic correspondence between rgb-images through aligning 3d object-class representations. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 1138–1147 (2025)
30. Wang, W., Weißberg, T., El Amrani, N., Bernard, F.: kh: Symmetry understanding of 3d shapes via chirality disentanglement. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 28292–28302 (2025)

31. Wang, Y., Luo, C.: Key-axis-based localization of symmetry axes in 3d objects utilizing geometry and texture. *IEEE Transactions on Image Processing* (2024)
32. Weißberg, T., Wang, W., Roetzer, P., El Amrani, N., Bernard, F.: Symmetry informative and agnostic feature disentanglement for 3d shapes. In: *Thirteenth International Conference on 3D Vision* (2026)
33. Yang, T., Rahman, M.A., Yeh, R.A.: Clipsym: Delving into symmetry detection with clip. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 21003–21013 (2025)
34. Yu, W., Seo, A., Cho, M.: Axis-level symmetry detection with group-equivariant representation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 24791–24800 (2025)
35. Zhang, J., Herrmann, C., Hur, J., Chen, E., Jampani, V., Sun, D., Yang, M.H.: Telling left from right: Identifying geometry-aware semantic correspondence. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3076–3085 (2024)
36. Zhang, J., Herrmann, C., Hur, J., Polania Cabrera, L., Jampani, V., Sun, D., Yang, M.H.: A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. In: *Advances in Neural Information Processing Systems*. vol. 36, pp. 45533–45547 (2023)
37. Zhu, J., Ju, Y., Zhang, J., Wang, M., Yuan, Z., Hu, K., Xu, H.: Densematcher: Learning 3d semantic correspondence for category-level manipulation from a single demo. In: *International Conference on Learning Representations*. vol. 2025, pp. 49538–49559 (2025)