
Symmetry-Robust 3D Orientation Estimation

Christopher Scarvelis¹ David Benhaim² Paul Zhang²

Abstract

Orientation estimation is a fundamental task in 3D shape analysis which consists of estimating a shape’s orientation axes: its side-, up-, and front-axes. Using this data, one can rotate a shape into canonical orientation, where its orientation axes are aligned with the coordinate axes. Developing an orientation algorithm that reliably estimates complete orientations of general shapes remains an open problem. We introduce a two-stage orientation pipeline that achieves state of the art performance on up-axis estimation and further demonstrate its efficacy on full-orientation estimation, where one seeks all three orientation axes. Unlike previous work, we train and evaluate our method on all of Shapenet rather than a subset of classes. We motivate our engineering contributions by theory describing fundamental obstacles to orientation estimation for rotationally-symmetric shapes, and show how our method avoids these obstacles.

1. Introduction

Orientation estimation is a fundamental task in 3D shape analysis which consists of estimating a shape’s orientation axes: its side-, up-, and front-axes. Using this data, one can rotate a shape into *canonical orientation*, in which the shape’s orientation axes are aligned with the coordinate axes. This task is especially important as a pre-processing step in 3D deep learning, where deep networks are typically trained on datasets of canonically oriented shapes but applied to arbitrarily-oriented shapes at inference time. While data augmentation or equivariant and invariant architectures may improve a model’s robustness to input rotations, these techniques come at the cost of data efficiency and model expressivity (Kuchnik & Smith, 2019; Kim et al., 2023). In contrast, orientation estimation allows one to pre-process shapes at inference time so that their orientation matches a model’s training data.

¹MIT CSAIL, Cambridge, MA ²Backflip AI, San Francisco, CA. Correspondence to: Christopher Scarvelis <scavr@mit.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

A shape’s orientation axes are not intrinsic geometric quantities: They are determined by humans based on physical and functional considerations. It is therefore difficult to construct simple geometric algorithms for orientation estimation. However, given a large dataset of canonically oriented shapes, one may pose orientation estimation as a supervised learning problem. This task is challenging, and developing an orientation pipeline that reliably estimates complete orientations of general shapes remains an open problem.

The naïve deep learning approach is to train a model with an L_2 loss to directly predict a shape’s orientation from a point cloud of surface samples. However, this strategy fails for shapes with rotational symmetries, where the optimal solution to the L_2 regression problem is the *Euclidean mean* (Moakher, 2002) of a shape’s orientations over all of its symmetries. In contrast, works such as Poursaeed et al. (2020) discretize the unit sphere into a set of fixed rotations and train a classifier to predict a probability distribution over these rotations, but find that this approach fails for any sufficiently dense discretization of the unit sphere.

Our key insight is to divide orientation estimation into two tractable sub-problems. In the first stage (the *quotient orienter*), we solve a continuous regression problem to recover a shape’s orientation *up to octahedral symmetries*. In the second stage (the *flipper*), we solve a discrete classification problem to predict one of 24 octahedral flips that returns the first-stage output to canonical orientation. Octahedral symmetries form a small set covering a substantial proportion of the symmetries occurring in real-world shapes. Consequently, quotienting our first-stage regression problem by octahedral symmetries prevents its predictions from collapsing to averages, while also keeping the subsequent classification problem tractable.

Using this strategy, our method accurately estimates a shape’s orientation up to one of its symmetries, which suffices for returning the shape to an upright and front-facing pose. As we are unaware of prior work on this problem, we also benchmark our pipeline’s performance on the well-studied task of *up-axis prediction*, in which one seeks to return a shape to an upright (but not necessarily front-facing) pose, and find that our work achieves state-of-the-art performance on this task. Unlike previous work, we train and

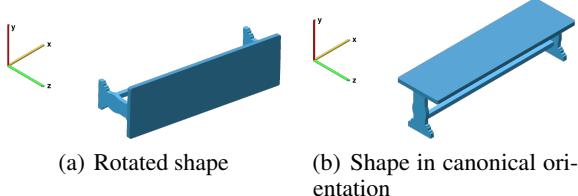


Figure 1: *Orientation estimation* allows users to rotate arbitrary shapes (a) into canonical orientation (b), in which the shape’s orientation axes are aligned with the coordinate axes.

evaluate our model on the *entire* Shapenet dataset rather than a subset of classes. We further demonstrate its generalization capabilities on Objaverse, a large dataset of real-world 3D models of varying quality.

A shape’s ground truth orientation may be ambiguous. This challenge is especially salient for nearly-symmetric shapes, where multiple orientations may yield nearly indistinguishable shapes. To resolve this issue, we use conformal prediction to enable our flipper to output *adaptive prediction sets* (Romano et al., 2020) whose size varies with the flipper model’s uncertainty. For applications with a human in the loop, this enables the end user to choose from a small set of plausible candidate orientations, dramatically simplifying the orientation estimation task while preserving user control over the outputs.

Our contributions include the following: (1) we identify fundamental obstacles to orientation estimation and study the conditions under which a naïve regression-based approach to orientation estimation fails; (2) we propose a two-stage orientation estimation pipeline that sidesteps these obstacles; (3) we train and test our model on Shapenet and show that it achieves SOTA performance for up-axis prediction and orientation estimation; (4) we use conformal prediction to enable end users to resolve ambiguities in a shape’s orientation; (5) we release our code and model weights to share our work with the ML community.

2. Related Work

Classical methods. A simple method for orientation estimation is to compute a rotation that aligns a shape’s principal axes with the coordinate axes; Kaye & Ivrissimtzis (2015) propose a robust variant of this method for mesh alignment. However, Kazhdan et al. (2003) find that PCA-based orientation estimation is not robust to asymmetries. Jin et al. (2012); Wang et al. (2014) propose unsupervised methods that leverage low-rank priors on axis-aligned 2D projections and third-order tensors, respectively, constructed from input shapes. These priors are restrictive, and the resulting orientation pipelines also fail on asymmetric shapes.

Another set of classical methods observe that as many man-made objects are designed to stand on flat surfaces, their up axis is normal to a *supporting base*. Motivated by this observation, these methods attempt to identify a shape’s supporting base rather than directly infer their up axis. Fu et al. (2008) generate a set of candidate bases, extract geometric features, and combine a random forest and SVM to predict a natural base from the candidates. Lin & Tai (2012) simplify a shape’s convex hull, cluster the resulting facets to obtain a set of candidate bases, and compute a hand-designed score to select the best base. Both of these methods rely heavily on feature engineering and fail on shapes that do not have natural supporting bases.

Deep learning-based methods. Motivated by the limitations of classical approaches, several works use deep learning for orientation estimation. Liu et al. (2016) train two neural networks on voxel representations of 3D shapes. A first-stage network assigns each shape to one of C classes. Based on this prediction, the shape is routed to one of C second-stage networks that are independently trained to predict the up axis from voxel representations of shapes in their respective classes. This method is unable to handle shapes that lie outside the C classes on which the networks were trained.

Pang et al. (2022) draw inspiration from classical methods and train a segmentation network to predict points that belong to a shape’s supporting base. They fit a plane to the predicted base points and output a normal vector to this plane as the predicted up axis. This method represents the current state of the art for orientation estimation, but struggles to handle shapes without well-defined natural bases and only predicts a shape’s up axis. In contrast, our method succeeds on general shapes and predicts a full rotation matrix that returns a shape to canonical orientation.

Chen et al. (2021) use reinforcement learning to train a model to gradually rotate a shape into upright orientation. While this algorithm performs well, it is evaluated on few classes and is costly to train. Kim et al. (2020) adopt a similar perspective to Fu et al. (2008), but use ConvNets to extract features for a random forest classifier that predicts a natural base. Poursaeed et al. (2020) use orientation estimation as a pretext task to learn features for shape classification and keypoint prediction. They also investigate a pure classification-based approach to orientation estimation that discretizes the 3D rotation group into K rotations and predicts a distribution over these rotations for an arbitrarily-rotated input shape. They find that its performance decays rapidly as K increases, reaching an accuracy as low as 1.6% for $K = 100$ rotations.

We also highlight a related literature on *canonical alignment*. This literature includes works such as Kim et al. (2023); Sajnani et al. (2022); Spezialetti et al. (2020); Zhou

et al. (2022), which seek to map arbitrarily-rotated shapes to a class-consistent pose, as well as Katzir et al. (2022); Sun et al. (2021), which seek to learn pose-invariant representations of 3D shapes. These works only attempt to learn a consistent orientation within each class, but this orientation is not consistent across classes and is not generally aligned with the coordinate axes. In contrast, we tackle the more challenging task of inferring a canonical orientation that is consistent across *all* objects.

More broadly, geometric deep learning studies general methods for exploiting symmetries in data by designing machine learning models that are invariant or equivariant to certain transformations. For instance, Kaba et al. (2023) proposes using learned canonicalization functions to obtain equivariant machine learning methods, and Puny et al. (2022) proposes frame averaging to induce invariance or equivariance in deep learning architectures. Our work may be viewed as an efficient canonicalization method for the specific case of 3D shapes with rotational symmetries.

3. Method

In this section, we motivate and describe our orientation pipeline. We first identify fundamental obstacles to orientation estimation and show that learning a shape’s orientation with the L_2 loss fails when the shape is rotationally symmetric. Motivated by these observations, we introduce our two-stage orientation pipeline consisting of a *quotient orienter* followed by a *flipper*. Our quotient orienter model solves a regression problem to recover a shape’s orientation up to octahedral symmetries, which commonly occur in real-world shapes. The flipper then predicts one of 24 octahedral flips that returns the first-stage output to canonical orientation. We finally use conformal prediction to enable our flipper to output prediction sets whose size varies with the model’s uncertainty. This allows end users to resolve ambiguities in a shape’s orientation by choosing from a small set of plausible candidate orientations.

3.1. Orientation estimation under rotational symmetries

In this section, we introduce the orientation estimation problem and motivate our approach. Throughout these preliminaries, we consider 3D shapes $S \in \mathcal{S}$ lying in some space of arbitrary shape representations \mathcal{S} . *Orientation estimation* consists of learning an *orienter* function $f : \mathcal{S} \rightarrow SO(3)$ that maps a shape $S \in \mathcal{S}$ to a predicted orientation $\hat{\Omega}_S \in SO(3)$, where $SO(3)$ denotes the 3D rotation group. An *orientation* is a rotation matrix Ω_S associated with a shape S that is rotation-equivariant: If one rotates S by $R \in SO(3)$ to obtain RS , then $\Omega_{RS} = R\Omega_S$. We interpret the columns of $\Omega_S = (\omega_S^x, \omega_S^y, \omega_S^z)$ as the side-, up-, and front-axes of S , respectively, and say that S is in *canonical orientation* if $\Omega_S = I$. If S is in canonical

orientation, then its side-, up-, and front-axes (which we will jointly refer to as its *orientation axes*) are aligned with the $\{x, y, z\}$ coordinate axes, respectively. We depict a canonically oriented shape S along with its orientation Ω_S in Figure 2.

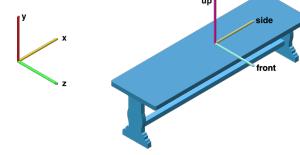


Figure 2: A shape’s *orientation* Ω_S is a rotation matrix whose columns are the shape’s side-, up-, and front-axes (plotted in yellow, magenta, cyan, resp.).

A shape’s orientation axes are not intrinsic geometric quantities. Instead, they are determined by humans based on physical and functional considerations. For instance, a shape’s up-axis may be determined by how a shape naturally lies under gravity, whereas its front-axis may be determined by the direction from which a human typically interacts with it. It is therefore challenging to devise simple geometric algorithms for orientation estimation.

However, there exist large datasets of shapes such as Shapenet (Chang et al., 2015) that have been assigned a canonical orientation by human annotators. Given a training set \mathcal{D} of shapes $S \in \mathcal{S}$ paired with their ground truth orientations Ω_S (where $\Omega_S \equiv I$ for datasets of canonically oriented shapes), a natural strategy for orientation estimation is to parametrize the orienter as a neural network f_θ and solve the following supervised learning problem:

$$\min_{f_\theta} \mathbb{E}_{\substack{R \sim U(SO(3)) \\ (S, \Omega_S) \in \mathcal{D}}} [\|f_\theta(RS) - R\Omega_S\|_F^2], \quad (1)$$

where $U(SO(3))$ is the uniform distribution over $SO(3)$.

Many real-world shapes S possess at least one non-trivial rotational symmetry R . This rotation leaves the shape unchanged, so $RS = S$. However, since a shape’s orientation is rotation-equivariant, $\Omega_{RS} = R\Omega_S \neq \Omega_S$, and the same shape S will necessarily be associated with two distinct orientations Ω_S and $R\Omega_S$. It follows that the ground truth orienter map $S \mapsto \Omega_S$ of a rotationally-symmetric shape is one-to-many and therefore not a function. We formally state and prove this result in Proposition A.1, and depict an instance of this phenomenon in Figure 3.

On the other hand, f_θ is a function, which implies that the solution to Equation 1 cannot be the true orienter map. The following proposition shows that for single shapes with rotational symmetries, the solution to Equation 1 is instead *Euclidean mean* of the rotated orientations $RQ\Omega_S$ across all rotations Q in the symmetry group \mathcal{R}_S (Moakher, 2002).

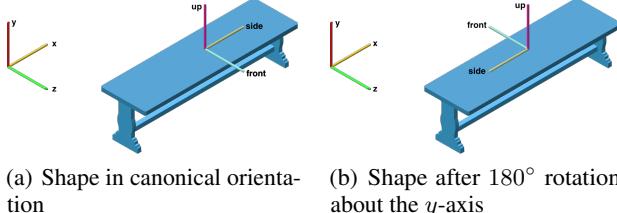


Figure 3: Rotating a shape by one of its symmetries changes its orientation while leaving the shape unchanged. Here, the front axis (in cyan) and side axis (in yellow) are flipped when the shape is rotated 180° about the y -axis.

Proposition 3.1. Let $S \in \mathcal{S}$ be a fixed shape which is symmetric under a non-trivial group of rotations $\mathcal{R}_S \subseteq SO(3)$. Let Ω_S be the shape’s orientation, and suppose $f^* : \mathcal{S} \rightarrow SO(3)$ solves the following regression problem:

$$\min_{f: \mathcal{S} \rightarrow SO(3)} \mathbb{E}_{R \sim U(SO(3))} [\|f(RS) - R\Omega_S\|_F^2], \quad (2)$$

Then $f^*(RS) = \text{proj}_{SO(3)} \left[\frac{1}{|\mathcal{R}_S|} \sum_{Q \in \mathcal{R}_S} RQ\Omega_S \right] \neq R\Omega_S$, where $\text{proj}_{SO(3)}$ denotes the orthogonal projection onto $SO(3)$.

We prove this proposition in Appendix A.2. This problem may be highly degenerate, even for shapes with a single non-trivial symmetry. For example, consider the bench shape S depicted in Figures 1, 2, 3. As shown in Figure 3, this shape has two rotational symmetries: The identity rotation, and a 180° rotation about the y -axis. It is straightforward to show that the solution f^* to Equation 2 is non-unique and may be any rotation about the y -axis, which we illustrate in Figure 4. (See Appendix A.3 for further details.) This demonstrates that even a single non-trivial rotational symmetry leads to *an entire submanifold of solutions* $f^*(S)$ to Problem 2, which substantiates the observations of Liu et al. (2016); Poursaeed et al. (2020) that orientation estimation via L_2 regression typically performs poorly in practice.

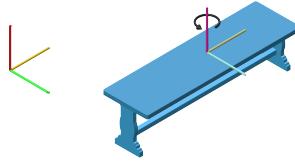


Figure 4: The solution $f^*(S)$ to Problem 2 evaluated at the bench shape S may be any rotation about the y -axis.

3.2. A partial solution.

The previous section shows that orientation estimation via L_2 regression fails for rotationally-symmetric shapes, which

are common in practice. We now present a partial solution to this problem. Suppose we know a finite group $\hat{\mathcal{R}} \supseteq \mathcal{R}_S$ that contains a shape S ’s rotational symmetries. We can then quotient the L_2 loss by $\hat{\mathcal{R}}$ to obtain the following problem:

$$\min_{f: \mathcal{S} \rightarrow SO(3)} \mathbb{E}_{R \sim U(SO(3))} \left[\min_{(S, \Omega_S) \in \mathcal{D}} \|f(RS) - RQ\Omega_S\|_F^2 \right]. \quad (3)$$

This loss is small if $f(RS)$ is close to the orientation $RQ\Omega_S$ of the rotated shape RQS for any $Q \in \hat{\mathcal{R}}$; Mehr et al. (2018) use similar techniques to learn latent shape representations that are invariant under a group of geometric transformations. Intuitively, whereas Equation 2 attempts to make $f(S)$ close to all $Q\Omega_S$, a minimizer of Equation 3 merely needs to make $f(S)$ close to *any* $Q\Omega_S$. Formally:

Proposition 3.2. Let $S \in \mathcal{S}$ be a fixed shape which is symmetric under a group of rotations $\mathcal{R}_S \subseteq \hat{\mathcal{R}} \subseteq SO(3)$. Let Ω_S be the shape’s orientation, and consider the following quotient regression problem:

$$\min_{f: \mathcal{S} \rightarrow SO(3)} \mathbb{E}_{R \sim U(SO(3))} \left[\min_{Q \in \hat{\mathcal{R}}} \|f(RS) - RQ\Omega_S\|_F^2 \right], \quad (4)$$

Then for any $R \in SO(3)$, there exists a solution $f^* : \mathcal{S} \rightarrow SO(3)$ of the form $f^*(RS) = RQ^*\Omega_S$ for some $Q^* \in \hat{\mathcal{R}}$.

We prove this proposition in Appendix A.4. In contrast to naïve L_2 regression, quotient regression learns a function that correctly orients rotationally-symmetric shapes *up to a rotation* in the group $\hat{\mathcal{R}}$. While this is only a partial solution to the orientation estimation problem, the remainder reduces to a discrete classification problem: Predicting the rotation $Q^* \in \hat{\mathcal{R}}$ such that $f^*(RS) = RQ^*\Omega_S$. In the following section, we will show how a solution to this problem allows one to map RS to the canonically oriented shape S .

3.3. Recovering an orientation via classification.

By solving the quotient regression problem in Equation 3, one can recover an arbitrarily-rotated shape RS ’s orientation up to a rotation $Q^* \in \hat{\mathcal{R}}$. In this section, we propose training a classifier to predict this rotation Q^* given the solution $f^*(RS) = RQ^*\Omega_S$ to the quotient regression problem. We now further assume that the shape S ’s ground truth orientation Ω_S is the canonical orientation $\Omega_S = I$. We show that even if S is symmetric under some group of symmetries $\mathcal{R}_S \subseteq \hat{\mathcal{R}}$, the optimal classifier’s predictions enable one to map RS to the canonically oriented shape S .

Predicting a rotation $Q^* \in \hat{\mathcal{R}}$ from the output $f^*(RS) = RQ^*$ of the quotient regression model is an $|\hat{\mathcal{R}}|$ -class classification problem. While one may hope that composing the

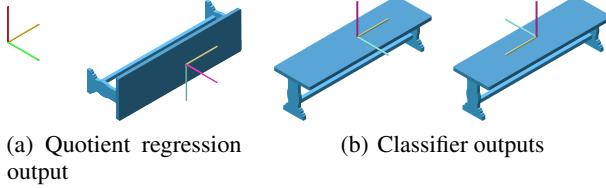


Figure 5: The quotient regression problem 4 correctly orients an arbitrarily rotated shape RS up to a rotation in $\hat{\mathcal{R}}$. The classification problem 5 then recovers the orientation of RS up to one of its rotational symmetries, which suffices for mapping RS to the canonically oriented shape S .

quotient regression model and the optimal classifier yields a function that outputs correct orientations $\hat{\Omega}_{RS} = R\Omega_S$ regardless of its inputs’ symmetries, such a function cannot exist for symmetric shapes. However, the following result shows that this pipeline recovers the orientation of a shape RS up to one of its rotational symmetries, which suffices for mapping RS to S .

Proposition 3.3. *Let $S \in \mathcal{S}$ be a fixed shape which is symmetric under a group of rotations $\mathcal{R}_S \subseteq \hat{\mathcal{R}} \subseteq SO(3)$, and suppose S is canonically oriented, so $\Omega_S = I$. Let $f^* : S \rightarrow SO(3)$ be a solution to Equation 3, so that $f^*(RS) = RQ^*$ for some $Q^* \in \hat{\mathcal{R}}$. Finally, suppose that $p^* : S \rightarrow \Delta^{|\hat{\mathcal{R}}|-1}$ solves the following problem:*

$$\min_{p: S \rightarrow \Delta^{|\hat{\mathcal{R}}|-1}} \mathbb{E}_{Q \sim U(\hat{\mathcal{R}})} [CE(p(QS), \delta_Q)], \quad (5)$$

where $U(\hat{\mathcal{R}})$ denotes the uniform distribution on $\hat{\mathcal{R}}$, $CE(\cdot)$ denotes the cross-entropy loss, and $\delta_Q \in \Delta^{|\hat{\mathcal{R}}|-1}$ is a one-hot vector centered at the index of $Q \in \hat{\mathcal{R}}$. Then for any $R \in SO(3)$, $p^*(f^*(RS)^\top RS)$ is the uniform distribution over $\{(Q^*)^\top F : F \in \mathcal{R}_S\}$. For any $(Q^*)^\top F$ in the support of this distribution, $\underbrace{((Q^*)^\top F)^\top}_{\text{second-stage prediction}} \underbrace{f^*(RS)^\top}_{\text{first-stage prediction}} RS = S$, so using f^* and p^* , one may recover S from the arbitrarily-rotated shape RS .

We prove this proposition in Appendix A.5. The results in this section show that unlike naïve L_2 regression, a two-stage pipeline consisting of quotient regression followed by discrete classification can successfully recover a shape’s orientation up to its symmetries, provided one quotients the L_2 objective by a sufficiently large subgroup of $SO(3)$. We combine these results in the following section to implement a state-of-the-art method for orientation estimation.

3.4. Implementation

Informed by our insights from Sections 3.2 and 3.3, we now present our state-of-the-art method for orientation estimation. Our pipeline consists of two components. Our first component, which we call the *quotient orienter*, is a neural network trained to solve Problem 3. We quotient the L_2 objective by $\hat{\mathcal{R}} := \mathcal{O} \subseteq SO(3)$, the *octahedral group* containing the 24 rotational symmetries of a cube. This is among the largest finite subgroups of $SO(3)$ (only the cyclic group C_n for $n \geq 48$ and dihedral group D_n for $n \geq 4$ can contain more subgroups), and it includes many rotational symmetries that commonly occur in real-world shapes.

Our second component, which we call the *flipper*, is a classifier trained to predict the rotation $Q^* \in \hat{\mathcal{R}}$ from the output $f^*(RS) = RQ^*$ of the quotient regression model. We illustrate the output of each stage of this pipeline in Figure 5. As many shapes possess multiple plausible orientations, we use conformal prediction to enable our flipper to output *adaptive prediction sets* whose size varies with the flipper model’s uncertainty. We provide further implementation details below.

Quotient orienter. We parametrize our quotient orienter by a DGCNN (Wang et al., 2019) operating on point clouds. To ensure that our predicted orientations lie in $SO(3)$, we follow Brégier (2021) and map model outputs from $\mathbb{R}^{3 \times 3}$ to $SO(3)$ by solving the special orthogonal Procrustes problem. We train the quotient orienter on point clouds sampled from the surfaces of meshes in Shapenet (Chang et al., 2015). As these meshes are pre-aligned to lie in canonical orientation, we fix $\Omega_S = I$ for all training shapes S . We provide full architecture and training details in Appendix B.

In our experiments, we observe that our quotient orienter yields accurate predictions for most input rotations R but fails for a small subset of rotations. To handle this, we follow Liu et al. (2016) and employ test-time augmentation to improve our model’s predictions. This consists of (1) randomly rotating the inputs RS by K random rotations $R_k \sim U(SO(3))$, $k = 1, \dots, K$, (2) obtaining the quotient orienter’s predictions $f_\theta(R_k RS)$ for each shape, (3) returning these predictions to the original input’s orientation by computing $R_k^\top f_\theta(R_k RS)$, and (4) outputting the prediction $R_{k^*}^\top f_\theta(R_{k^*} RS)$ with the smallest average quotient distance to the remaining predictions.

Flipper. We also parametrize our flipper by a DGCNN operating on point clouds. We train the flipper on point clouds sampled from the surface of Shapenet meshes by solving the $|\hat{\mathcal{R}}|$ -class classification problem described in Section 3.3. We draw rotations $Q \sim U(\mathcal{O})$ during training, and simulate inaccuracies in our quotient orienter’s predictions by further rotating the training shapes about a randomly drawn axis by

an angle uniformly drawn from $[0, 10]$ degrees. We provide full architecture and training details in Appendix B.

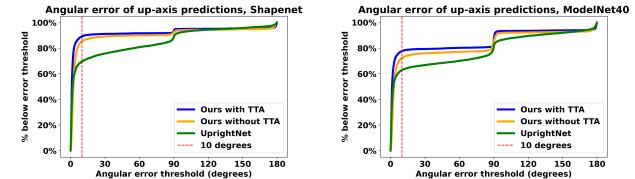
We also employ test-time augmentation (TTA) to improve our flipper model’s predictions. Similarly to the case with the quotient orienter, we (1) randomly flip the inputs by K random rotations $R_k \sim \hat{\mathcal{R}} = \mathcal{O}$, (2) obtain the flipper’s predictions for each shape, (3) return these predictions to the original input’s orientation, and (4) output the plurality prediction.

Adaptive prediction sets. Many real-world shapes have several plausible canonical orientations, even when they lack rotational symmetries. Furthermore, the flipper model may map nearly-symmetric shapes with unique canonical orientations to a uniform distribution over their near-symmetries due to factors such as insufficiently dense point clouds or the smoothness of the flipper function.

To mitigate this issue in pipelines with a human in the loop, we enable our flipper model to output *adaptive prediction sets* whose size varies with the flipper’s uncertainty (Romano et al., 2020). This method uses a small *conformal calibration set* drawn from the validation data to learn a threshold parameter $\tau > 0$ associated with a coverage probability $\alpha \in (0, 1)$ that controls the size of the prediction sets. Given the flipper model’s output probabilities $p_\phi(S) \in \Delta^{|\hat{\mathcal{R}}|-1}$ for some shape S , one sorts $p_\phi(S)$ in descending order and adds elements of $\hat{\mathcal{R}}$ to the prediction set until their total mass in $p_\phi(S)$ reaches τ . Intuitively, these sets will be small when the flipper is confident in its prediction and assigns large mass to the highest-probability classes. Conversely, the sets will be large when the flipper is uncertain and assigns similar mass to most classes.

4. Experiments

We now evaluate our method’s performance on orientation estimation. We first follow the evaluation procedure in Pang et al. (2022) and benchmark against their “Upright-Net,” which represents the current state of the art for orientation estimation. Upright-Net can only map shapes into upright orientation, where a shape’s up-axis is aligned with the y -axis; in contrast, our method recovers a full orientation Ω_S for each shape. We therefore follow this benchmark with an evaluation of our method on the more challenging task of full-orientation estimation. We incorporate adaptive prediction sets at this stage and demonstrate that our method reliably provides a plausible set of candidate orientations for diverse shapes unseen during training. We train and evaluate all models on Shapenet (Chang et al., 2015), as this is the largest and most diverse dataset we are aware of containing canonically oriented shapes. However, we report qualitative results for our method’s out-of-distribution performance on Objaverse in Appendix C.



(a) Angular error histograms for Shapenet

(b) Angular error histograms for ModelNet40

Figure 6: Comparison of angular errors between the estimated and ground truth up-axis on the Shapenet validation set (left) and on ModelNet40 (right). We plot the empirical CDF of the angular errors of each model’s outputs. The dashed lines indicate the 10° error threshold beyond which a prediction is treated as incorrect. With test-time augmentation (TTA), our algorithm’s error rate is 64.6% lower than the prior state of the art.

4.1. Upright orientation estimation

We construct a random 90-10 train-test split of Shapenet, draw 10k point samples from the surface of each mesh, and train our quotient orienter and flipper on all classes in the training split. We train our quotient orienter for 1919 epochs and our flipper for 3719 epochs, sampling 2k points per point cloud at each iteration and fixing a learning rate of 10^{-4} . We also train Upright-Net with 2048 points per cloud on the same data for 969 epochs at the same learning rate, at which point the validation accuracy has plateaued. We follow the annotation procedure in Pang et al. (2022) to obtain ground truth segmentations of each point cloud into supporting base points and non-base points.

Table 1: Up-axis estimation accuracy for our pipeline trained on Shapenet

Method	Accuracy (\uparrow)	
	Shapenet	ModelNet40
Ours (with TTA)	89.2 %	77.7 %
Ours (without TTA)	85.3 %	72.3 %
Upright-Net (Pang et al., 2022)	69.5 %	62.3 %

We then follow the evaluation procedure in Pang et al. (2022) to benchmark our method against their SOTA method for upright orientation estimation. We randomly rotate shapes S in the validation set, use our two-stage pipeline and Upright-Net to estimate the up-axis ω_{RS}^y of each randomly rotated shape RS , and then measure the *angular error* $\arccos(\langle \hat{\omega}_{RS}^y, \omega_{RS}^y \rangle)$ between the estimated and ground truth up-axis. Our method’s estimated up-axis is the second column of our estimated orientation matrix $\hat{\Omega}_{RS}$.

In contrast, Upright-Net predicts a set of base points for RS , fits a plane to these points, and returns this plane’s normal

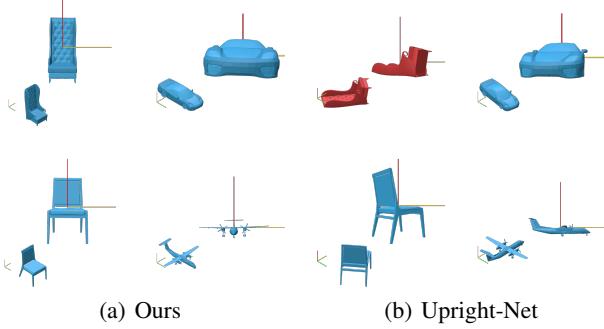


Figure 7: Comparison of oriented shapes recovered from randomly rotated inputs using our algorithm (left) and Upright-Net (right). Failures are rendered in red. Our algorithm recovers correct upright and front-facing orientations for most shapes, whereas Upright-Net cannot recover front-facing orientations and fails over $2.8\times$ as often at up-axis prediction.

vector pointing towards the shape’s center of mass. This method relies on a restrictive prior on the geometry of the input shapes that does not hold for shapes which do not naturally lie on a supporting base. We follow Pang et al. (2022) and define our methods’ respective accuracies to be the proportion of validation meshes whose angular error is less than 10° .

We depict the results of this benchmark in Table 1. **Our method improves on Upright-Net’s up-axis estimation accuracy by nearly 20 percentage points**, corresponding to a 64.6% reduction in the error rate relative to the previous state of the art. To provide a more comprehensive picture of our respective models’ performance, we also report the empirical CDF of angular errors for our model and Upright-Net in the left panel of Figure 6. Our model primarily fails by outputting orientations that are 90° or 180° away from the correct orientation, which correspond to failures of the flipper. In contrast, Upright-Net’s failures are more evenly distributed across angular errors. Finally, we depict a grid of non-cherry-picked outputs of our model and Upright-Net in Figure 7 and highlight failure cases in red.

We quantitatively evaluate our model’s generalization by performing the same experiment on ModelNet40 (Wu et al., 2015). Both models’ performances deteriorate in this setting, but our algorithm continues to substantially outperform Upright-Net. Furthermore, the right panel of Figure 6 shows that our model’s failures on ModelNet40 are more heavily weighted towards flipper failures (where the angular error is close to 90° and 180°). In the following section, we will show how a human in the loop can resolve these failures by choosing from a small set of candidate flips, which substantially improves our pipeline’s quantitative performance.

These results demonstrate that our method significantly im-

proves over the state of the art in up-axis estimation. In the following section, we show that our method also successfully recovers the full orientation Ω_{RS} of a rotated shape up to its symmetries, a more challenging task than upright orientation estimation. Using our estimated orientations, we return a wide variety of shapes into canonical orientation.

4.2. Full-orientation estimation

We now evaluate our method’s performance on full-orientation estimation, in which we use our model’s full predicted orientation matrix $\hat{\Omega}_{RS}$ to transform an arbitrarily-rotated shape RS to the canonically oriented shape S . We draw shapes S from the Shapenet validation set and randomly rotate them to obtain inputs RS to our pipeline. Because several orientation matrices may be associated with a single canonically oriented shape when this shape has rotational symmetries, we report the *symmetric chamfer distance* between our pipeline’s output shape and the ground truth shape. Because this is a shape-to-shape metric, it is insensitive to rotational symmetries: If $RS = S$, then $d(S, T) = d(RS, T)$, where d is the chamfer distance and T is a reference shape.

To our knowledge, our algorithm is the first to solve this task for generic shapes without requiring class information at training time or at inference time. We therefore construct two baselines using Upright-Net. As this algorithm only predicts a shape’s up-axis (the second column of its orientation matrix), we must augment Upright-Net’s predictions with a front-axis and a side-axis. We generate a baseline called “Upright-Net-Oracle” by augmenting Upright-Net’s predicted up-axis with the shape’s ground truth front-axis, and a second baseline called “Upright-Net-Random” by augmenting the predicted up-axis with a random vector that we orthogonalize with respect to the up-axis. In both cases, we obtain the side-axis by taking the cross product of the respective up- and front-axes. “Upright-Net-Oracle” upper-bounds the performance of a hypothetical orienter built on Upright-Net, whereas “Upright-Net-Random” provides a lower bound on such an orienter’s performance.

We benchmark our pipeline’s performance against each baseline in Table 2, where we report the mean chamfer distance between each model’s predicted shapes and the corresponding ground truth shapes, which are in canonical orientation. Our method achieves an 84% reduction in mean chamfer distance between predicted and ground truth shapes relative to the *oracle* baseline, which augments Upright-Net’s predicted up-axis with the shape’s *ground truth* front-axis. In practice, one does not have access to this front-axis at inference time, so this represents a loose upper bound on the potential performance of a shape orientation algorithm built upon Upright-Net’s predicted up-axes.

This empirically validates our Proposition 3.2 and demon-

strates that our pipeline successfully recovers canonically oriented shapes, even when symmetries preclude the orienter map from being represented by a function. We further illustrate our pipeline’s performance in Figure 10 in Appendix C, where we depict a large grid of shapes that have been oriented by our pipeline. Even when our model’s outputs disagree with the ground truth pose in Shapenet (we render the corresponding meshes in red), the recovered shape is often acceptable in practice.

Finally, in Figure 12 in Appendix C, we depict transformed shapes obtained by applying our method to randomly-rotated shapes from the Objaverse dataset (Deitke et al., 2023). This dataset contains highly diverse meshes of varying quality and therefore serves as a useful test case for our method’s performance on out-of-distribution shapes. (As these shapes are not canonically oriented, we cannot train on them or report meaningful error metrics.) Using our orientation pipeline, one reliably recovers shapes that are canonically oriented up to an octahedral flip. Our flipper has greater difficulty handling out-of-distribution meshes, but predicts an acceptable flip in many cases. We expect that training our flipper on a larger dataset of oriented shapes will further improve its generalization performance.

Table 2: Full-orientation estimation performance on the Shapenet validation set.

Method	Mean chamfer distance (\downarrow)
Upright-Net-Random	0.10801 ± 0.13824
Upright-Net-Oracle	0.05481 ± 0.13016
Ours (TTA)	0.00856 ± 0.0396
Ours (w/o TTA)	0.01107 ± 0.04342
Ours (with APS)	0.00208 ± 0.01407

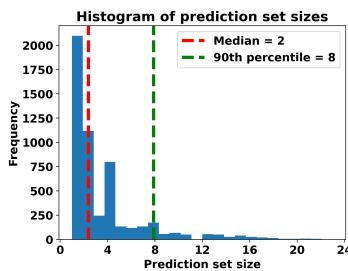


Figure 8: Histogram of APS sizes with coverage probability $\alpha = 0.3$. The median APS contains 2 shapes, and 90% of APSs contain at most 8 shapes.

Adaptive prediction sets. Many real-world shapes have several plausible canonical orientations, even if they lack rotational symmetries. In particular, while most real-world shapes have a well-defined upright orientation, their front-facing orientation is often ambiguous. In Section 3.4, we proposed enabling a human in the loop to resolve these am-

biguities by having our orientation pipeline output *adaptive prediction sets* (APSs), whose size varies with the flipper’s uncertainty. We now incorporate this technique into our evaluation. We learn a threshold parameter τ corresponding to a coverage probability of $\alpha = 0.3$, and at inference time output a *set* of flips in descending order of probability under the flipper’s output distribution until their total probability mass reaches τ . Intuitively, this set will be small when the flipper is confident in its prediction and large otherwise.

By applying each flip to the first-stage orienter’s output, we obtain a shape orientation pipeline that outputs *sets* of candidate shapes \hat{S}_c given a single arbitrarily-rotated input RS . To evaluate its performance, we compute the chamfer distance between each candidate shape and the ground truth shape, and take the *minimum* over these chamfer distances. As one may trivially reduce the minimum chamfer distance over an APS by outputting arbitrarily large sets, we report a histogram of APS sizes in Figure 8 to demonstrate that these sets are typically small.

We report our pipeline’s performance with adaptive prediction sets in the last row of Table 2. Allowing our pipeline to output an APS reduces the mean chamfer distance between the closest shape in this set to the ground truth by a factor of 4 relative to Orient Anything without this feature. Figure 8 shows that with our chosen coverage probability, most APSs are small; the median APS size is 2, and 90% of APSs have at most 8 shapes. This demonstrates that our flipper typically assigns most of its probability mass to a few octahedral symmetries, which often contain a rotation that returns the shape to its canonical pose in Shapenet. In Figure 11 in Appendix C, we also include a grid of meshes in the prediction sets associated with 20 randomly-selected validation shapes. These prediction sets typically consist of a small collection of plausible shape orientations; by manually inspecting them, a human in the loop may select the orientation that is most suitable for their target application.

5. Conclusion

This work introduces a state-of-the-art method for 3D orientation estimation. Whereas previous approaches can only infer upright orientations for limited classes of shapes, our method successfully recovers entire orientations for general shapes. We show that naïve regression-based approaches for orientation estimation degenerate on rotationally-symmetric shapes, which are common in practice, and develop a two-stage orientation pipeline that avoids these obstacles. Our pipeline first orients an arbitrarily rotated input shape up to an octahedral symmetry, and then predicts the octahedral symmetry that maps the first-stage output to the canonically oriented shape. We anticipate that this factorization of geometric learning problems will be broadly applicable throughout 3D deep learning for tackling problems that are ill-posed due to the presence of symmetries.

Acknowledgments

We thank the authors of Pang et al. (2022) for their gracious assistance with their source code and Justin Solomon for valuable discussions. Chris thanks Backflip AI for hosting him and generously supporting him with mentorship and compute throughout his internship, and gratefully acknowledges the support of a 2024 Exponent fellowship.

Impact statement

This paper presents a novel algorithm for 3D shape orientation. We do not believe that there are notable ethical considerations or societal impacts of our work meriting specific consideration.

References

- Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.
- Brégier, R. Deep regression on manifolds: a 3d rotation case study. In *2021 International Conference on 3D Vision (3DV)*, pp. 166–174. IEEE, 2021.
- Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.
- Chen, L., Xu, J., Wang, C., Huang, H., Huang, H., and Hu, R. Uprightrl: upright orientation estimation of 3d shapes via reinforcement learning. In *Computer Graphics Forum*, volume 40, pp. 265–275. Wiley Online Library, 2021.
- Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., and Farhadi, A. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023.
- Fu, H., Cohen-Or, D., Dror, G., and Sheffer, A. Upright orientation of man-made objects. In *ACM SIGGRAPH 2008 Papers, SIGGRAPH ’08*, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781450301121. doi: 10.1145/1399504.1360641. URL <https://doi.org/10.1145/1399504.1360641>.
- Gower, J. C. and Dijksterhuis, G. B. *Procrustes problems*, volume 30. OUP Oxford, 2004.
- Jin, Y., Wu, Q., and Liu, L. Unsupervised upright orientation of man-made models. *Graphical Models*, 74(4):99–108, 2012.
- Kaba, S.-O., Mondal, A. K., Zhang, Y., Bengio, Y., and Ravanbakhsh, S. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pp. 15546–15566. PMLR, 2023.
- Katzir, O., Lischinski, D., and Cohen-Or, D. Shape-pose disentanglement using se(3)-equivariant vector neurons. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pp. 468–484, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-20061-8. doi: 10.1007/978-3-031-20062-5_27. URL https://doi.org/10.1007/978-3-031-20062-5_27.
- Kaye, D. and Ivriissimtzis, I. Mesh alignment using grid based pca. In *International Conference on Computer Graphics Theory and Applications*, volume 2, pp. 174–181. SCITEPRESS, 2015.
- Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing*, volume 6, pp. 156–164, 2003.
- Kim, S., Park, C., Jeong, Y., Park, J., and Cho, M. Stable and consistent prediction of 3d characteristic orientation via invariant residual learning. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org, 2023.
- Kim, S.-h., Hwang, Y., and Kweon, I. S. Category-specific upright orientation estimation for 3d model classification and retrieval. *Image and Vision Computing*, 96:103900, 2020.
- Kuchnik, M. and Smith, V. Efficient augmentation via data subsampling. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Byxpfh0cFm>.
- Lin, C.-K. and Tai, W.-K. Automatic upright orientation and good view recognition for 3d man-made models. *Pattern recognition*, 45(4):1524–1530, 2012.
- Liu, Z., Zhang, J., and Liu, L. Upright orientation of 3d shapes with convolutional networks. *Graphical Models*, 85:22–29, 2016.
- Mehr, E., Lieutier, A., Bermudez, F. S., Guitteny, V., Thome, N., and Cord, M. Manifold learning in quotient spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9165–9174, 2018.

- Moakher, M. Means and averaging in the group of rotations. *SIAM journal on matrix analysis and applications*, 24(1):1–16, 2002.
- Pang, X., Li, F., Ding, N., and Zhong, X. Upright-net: Learning upright orientation for 3d point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14911–14919, 2022.
- Poursaeed, O., Jiang, T., Qiao, H., Xu, N., and Kim, V. G. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pp. 1018–1028. IEEE, 2020.
- Puny, O., Atzmon, M., Smith, E. J., Misra, I., Grover, A., Ben-Hamu, H., and Lipman, Y. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=zIUyj55nXR>.
- Romano, Y., Sesia, M., and Candes, E. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.
- Sajnani, R., Poulenard, A., Jain, J., Dua, R., Guibas, L. J., and Sridhar, S. Condor: Self-supervised canonicalization of 3d pose for partial shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16969–16979, 2022.
- Spezialetti, R., Stella, F., Marcon, M., Silva, L., Salti, S., and Di Stefano, L. Learning to orient surfaces by self-supervised spherical cnns. *Advances in Neural information processing systems*, 33:5381–5392, 2020.
- Sun, W., Tagliasacchi, A., Deng, B., Sabour, S., Yazdani, S., Hinton, G. E., and Yi, K. M. Canonical capsules: Self-supervised capsules in canonical pose. *Advances in Neural information processing systems*, 34:24993–25005, 2021.
- Wang, W., Liu, X., and Liu, L. Upright orientation of 3d shapes via tensor rank minimization. *Journal of Mechanical Science and Technology*, 28:2469–2477, 2014.
- Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.
- Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.
- Zhou, K., Bhatnagar, B. L., Schiele, B., and Pons-Moll, G. Adjoint rigid transform network: Task-conditioned alignment of 3d shapes. In *2022 international conference on 3D vision (3DV)*, pp. 1–11. IEEE, 2022.

A. Additional results and proofs

A.1. The orienter map for symmetric shapes is not a function

Proposition A.1. Let $S \in \mathcal{S}$ be a fixed shape which is symmetric under a non-trivial group of rotations $\mathcal{R}_S \subseteq SO(3)$, and let Ω_S be its orientation. Then there is no function f such that $f(RS) = R\Omega_S$ for all $R \in SO(3)$.

Proof: Suppose $f(RS) = R\Omega_S$ for all $R \in SO(3)$, and fix some non-identity rotation $R \in \mathcal{R}_S$ under which S is symmetric. Then $RS = S$, but $f(RS) = R\Omega_S \neq \Omega_S = f(S)$, so $f(RS) \neq f(S)$ even though $RS = S$. Hence f must be a one-to-many map and is therefore not a function. ■

A.2. Proposition 3.1

The key insight is that if f is a function, then $f(RS) = f(R'S)$ for all $R' \in SO(3)$ such that $RS = R'S$. Equation 2 will then drive the optimal $f^*(RS)$ to the *Euclidean mean* (Moakher, 2002) of the rotation matrices R' such that $RS = R'S$. We begin by showing that these are precisely the matrices RQ for $Q \in \mathcal{R}_S$.

As \mathcal{R}_S is the group of symmetries of S , $QS = S$ for all $Q \in \mathcal{R}_S$. Given some rotation $R \in SO(3)$, left-multiplying by R then yields $RQS = RS$ for all $Q \in \mathcal{R}_S$. This relationship also holds in reverse: If $RS = R'S$ for $R, R' \in SO(3)$, then $R' = RQ$ for some $Q \in \mathcal{R}_S$. To see this, note that if $RS = R'S$, then $S = R^\top R'S$ and hence $R^\top R' \in \mathcal{R}_S$. Consequently, $R' = R(R^\top R') = RQ$ for $Q := R^\top R' \in \mathcal{R}_S$. It follows that:

$$\{R' \in SO(3) : RS = R'S\} = \{RQ : Q \in \mathcal{R}_S\}.$$

We can therefore write a solution to Equation 2 evaluated at RS as follows:

$$\begin{aligned} f^*(RS) &= \underset{R^* \in SO(3)}{\operatorname{argmin}} \underset{R' \in SO(3): RS = R'S}{\mathbb{E}} [\|R^* - R'\Omega_S\|_F^2] \\ &= \underset{R^* \in SO(3)}{\operatorname{argmin}} \underset{RQ: Q \in \mathcal{R}_S}{\mathbb{E}} [\|R^* - RQ\Omega_S\|_F^2] \\ &= \underset{R^* \in SO(3)}{\operatorname{argmin}} \underset{Q \in U(\mathcal{R}_S)}{\mathbb{E}} [\|R^* - RQ\Omega_S\|_F^2] \\ &= \underset{R^* \in SO(3)}{\operatorname{argmin}} \frac{1}{|\mathcal{R}_S|} \sum_{Q \in \mathcal{R}_S} \|R^* - RQ\Omega_S\|_F^2. \end{aligned}$$

This is the Euclidean mean of the matrices $RQ\Omega_S$ as defined in Moakher (2002). Proposition 3.3 in the same reference states that the solution to this problem is found by computing the arithmetic mean $\frac{1}{|\mathcal{R}_S|} \sum_{Q \in \mathcal{R}_S} RQ\Omega_S$ and then orthogonally projecting this onto $SO(3)$. In particular,

$$f^*(RS) = \operatorname{proj}_{SO(3)} \left[\frac{1}{|\mathcal{R}_S|} \sum_{Q \in \mathcal{R}_S} RQ\Omega_S \right] \neq R\Omega_S.$$

Hence L_2 regression fails to learn the orientation Ω_S of a shape $S \in \mathcal{S}$ that possesses a non-trivial set of rotational symmetries \mathcal{R}_S . ■

A.3. Non-uniqueness of solution to Problem 2

Consider the bench shape S depicted in Figures 1, 2, 3. As shown in Figure 3, this shape has two rotational symmetries: The identity rotation, and a 180° rotation about the y -axis. One may represent these rotations by the matrices I and $Q := (-e_x, e_y, -e_z)$, respectively, where e_x, e_y, e_z are the standard basis vectors.

Proposition 3.1 states that one solves the L^2 regression problem 2 for the bench shape by computing the arithmetic mean of I and Q and then orthogonally projecting this matrix onto $SO(3)$. The arithmetic mean of I, Q is the matrix $M := (0, e_y, 0)$,

and one computes its orthogonal projection onto $SO(3)$ by solving a special Procrustes problem (Gower & Dijksterhuis, 2004):

$$\min_{R \in SO(3)} \|R - M\|_F^2 \quad (6)$$

The solution to this problem is non-unique for $M := (0, e_y, 0)$, and the minimum is attained by any rotation about the y -axis (i.e. any rotation matrix whose second column is e_y). This shows that even a single non-trivial rotational symmetry leads to *an entire submanifold of solutions* $f^*(S)$ to Problem 2.

A.4. Proposition 3.2

We begin by defining an equivalence relation over $SO(3)$. Given two rotations $R_1, R_2 \in SO(3)$, we call R_1, R_2 equivalent and write $R_1 \sim R_2$ if there exists some $Q \in \hat{\mathcal{R}}$ such that $R_2 = R_1 Q$. We verify that this is an equivalence relation:

Reflexivity: $I \in \hat{\mathcal{R}}$ since $\hat{\mathcal{R}}$ is a group and $R_1 = R_1 I$, so $R_1 \sim R_1$.

Symmetry: Suppose $R_1 \sim R_2$. Then $R_2 = R_1 Q$ for some $Q \in \hat{\mathcal{R}}$. As $\hat{\mathcal{R}}$ is a group, $R^\top = R^{-1} \in \hat{\mathcal{R}}$ as well, and $R_2 Q^\top = R_1$, so $R_2 \sim R_1$.

Transitivity: Suppose $R_1 \sim R_2$ and $R_2 \sim R_3$. Then there are $Q, Q' \in \hat{\mathcal{R}}$ such that $R_2 = R_1 Q$ and $R_3 = R_2 Q'$. Hence $R_3 = R_2 Q' = R_1 Q Q'$, and as $\hat{\mathcal{R}}$ is a group, $Q Q' \in \hat{\mathcal{R}}$. We conclude that $R_1 \sim R_3$.

This confirms that \sim is a valid equivalence relation. Using this equivalence relation, we partition $SO(3)$ into equivalence classes, choose a unique representative for each class, and use $[R] \in SO(3)/\sim$ to denote the unique representative for the equivalence class containing $R \in SO(3)$. We then use this map to define a candidate solution to Equation 4 over the space of rotated shapes $\{RS : R \in SO(3)\}$ as $f^*(RS) := [R]\Omega_S$. We will first verify that this defines a valid function (i.e. that f^* is not one-to-many), and then show that it attains a loss value of 0 in Equation 4.

We first show that f^* defines a valid function. To do so, we must show that if $R_1 S = R_2 S$, then $f^*(R_1 S) = f^*(R_2 S)$. To this end, suppose that $R_1 S = R_2 S$. Then $S = R_1^\top R_2 S$, so $Q := R_1^\top R_2 \in \mathcal{R}_S \subseteq \hat{\mathcal{R}}$. It follows that $R_2 = R_1 R_1^\top R_2 = R_1 Q$ for some $Q \in \hat{\mathcal{R}}$, so $R_1 \sim R_2$. Since $R_1 \sim R_2$, $[R_1] = [R_2]$ and so $f^*(R_1 S) = [R_1]\Omega_S = [R_2]\Omega_S = f^*(R_2 S)$. This shows that f^* defines a valid function.

We now show that f^* attains a loss value of 0 in Equation 4. For any $R \in SO(3)$, we have:

$$\min_{Q \in \hat{\mathcal{R}}} \|f(RS) - RQ\Omega_S\|_F^2 = \min_{Q \in \hat{\mathcal{R}}} \|[R]\Omega_S - RQ\Omega_S\|_F^2.$$

But clearly $R \sim [R]$, so there exists some $Q^* \in \hat{\mathcal{R}}$ such that $[R] = RQ^*$. Hence

$$\min_{Q \in \hat{\mathcal{R}}} \|[R]\Omega_S - RQ\Omega_S\|_F^2 = 0,$$

and as this reasoning holds for any $R \in SO(3)$, it follows that

$$\mathbb{E}_{R \sim U(SO(3))} \left[\min_{Q \in \hat{\mathcal{R}}} \|f(RS) - RQ\Omega_S\|_F^2 \right] = 0.$$

We conclude that f^* is a minimizer of Equation 4. Furthermore, $f^*(S) = [I]\Omega_S = Q^*\Omega_S$ for some $Q^* \in \hat{\mathcal{R}}$, which completes the proof of the proposition. ■

A.5. Proposition 3.3

If $F \in \mathcal{R}_S$, then $FS = S$, so $QFS = QS$ for any other rotation $Q \in SO(3)$ and $\{QFS : F \in \mathcal{R}_S\}$ contains the symmetries of the rotated shape QS . The optimal solution p^* to Equation 5 maps a rotated shape QS (where $Q \in \hat{\mathcal{R}}$) to the

empirical distribution of the targets $Q \in \hat{\mathcal{R}}$ conditional on a shape QS . But if $QFS = QS$ for all $F \in \mathcal{R}_S$, then this is the uniform distribution over the set $\{QF : F \in \mathcal{R}_S\}$.

Since $f^*(RS) = RQ^*$ for some $Q^* \in \hat{\mathcal{R}}$, $f^*(RS)^\top RS = (Q^*)^\top S$, and applying the general result from above, we conclude that $p^*(f^*(RS)^\top RS) = p^*((Q^*)^\top S)$ is the uniform distribution over the set $\{(Q^*)^\top F : F \in \mathcal{R}_S\}$.

For any $(Q^*)^\top F$, one then computes $((Q^*)^\top F)^\top f^*(RS)^\top RS = F^\top S$. But as \mathcal{R}_S is a group, $F^\top \in \mathcal{R}_S$ whenever F is, so $F^\top S = S$ and we conclude that $((Q^*)^\top F)^\top f^*(RS)^\top RS = S$. ■

B. Implementation details

B.1. Quotient orienter

We parametrize our quotient orienter by a DGCNN and use the author’s Pytorch implementation (Wang et al., 2019) with 1024-dimensional embeddings, $k = 20$ neighbors for the EdgeConv layers, and a dropout probability of 0.5. Our DGCNN outputs unstructured 3×3 matrices, which we then project onto $SO(3)$ by solving a special orthogonal Procrustes problem; we use the `roma` package (Brégier, 2021) to efficiently compute this projection.

We train our quotient orienter on point clouds consisting of 10k surface samples from Shapenet meshes. We subsample 2k points per training iteration and pass batches of 48 point clouds per iteration. We train the quotient orienter for 1919 epochs at a learning rate of 10^{-4} .

Test-time augmentation for the quotient orienter. Our use of TTA for the quotient orienter is motivated by the observation that the quotient orienter succeeds on most rotations and fails on small subsets of rotations. Given an arbitrarily-oriented input shape RS , we would like to mitigate the possibility that the shape is in an orientation for which the quotient orienter fails. To do so, we apply K random rotations R_k to the input shape to obtain randomly re-rotated shapes $R_k RS$. Because the quotient orienter only fails on small subsets of rotations, we expect it to succeed on most of the $R_k RS$ and recover the orientations of $R_k RS$ up to an octahedral symmetry.

Because we are interested in the orientation of RS rather than $R_k RS$, we then need to apply the inverse of R_k to each predicted orientation $f_\theta(R_k RS)$ to obtain a set of K predicted orientations $R_k^\top f_\theta(R_k RS)$. These will be correct orientations of RS up to octahedral symmetries for each k where the quotient orienter succeeds.

Because we expect this to be the case for most k , we employ a voting scheme to select one of the candidate orientations $R_k^\top f_\theta(R_k RS)$. In this step, we compute each candidate’s average quotient L_2 loss with respect to every other candidate (i.e. the loss in Problem 3), and choose the candidate which minimizes this measure. Because we expect the quotient orienter to have succeeded for most of the $R_k RS$, this average quotient loss will be small for most candidates and large for the few outlier candidates on which the orienter failed. Choosing the candidate with the minimum average quotient loss with respect to the other candidates filters out these outliers and makes it likelier that we output the correct orientation of RS up to an octahedral symmetry.

B.2. Flipper

We parametrize our flipper by a DGCNN and use the author’s Pytorch implementation (Wang et al., 2019) with 1024-dimensional embeddings, $k = 20$ neighbors for the EdgeConv layers, and a dropout probability of 0.5. Our flipper outputs 24-dimensional logits, as we quotient our first-stage regression problem by the octahedral group, which contains the 24 rotational symmetries of a cube.

We train our flipper on point clouds consisting of 10k surface samples from Shapenet meshes. We subsample 2k points per training iteration and pass batches of 48 point clouds per iteration. We train the quotient orienter for 3719 epochs at a learning rate of 10^{-4} . We draw rotations $Q \in U(\mathcal{O})$ during training, and simulate inaccuracies in our quotient orienter’s predictions by further rotating the training shapes about a randomly drawn axis by an angle uniformly drawn from $[0, 10]$ degrees.

Test-time augmentation for the flipper. The TTA procedure for the flipper is similar to the quotient orienter’s TTA procedure. We are given an input shape FS , which is correctly oriented up to an octahedral symmetry (a “flip”) F if the quotient orienter succeeded. We apply K random flips F_k to the input shape to obtain randomly re-rotated shapes $F_k FS$ and apply the flipper to each of these shapes. Similarly to the quotient orienter, if the flipper succeeds on some $F_k FS$, it

predicts $F_k F$ rather than the flip F we are actually interested in. We consequently left-multiply each prediction $g(F_k FS)$ by F_k^\top , which maps each successful prediction $g(F_k FS) = F_k F$ to the true flip F . Because we expect the flipper to have succeeded on most inputs $F_k FS$ and failed on a minority of inputs, we again use a voting scheme to pick out the plurality prediction. In this case, we simply return the most common flip among the set of $F_k^\top g(F_k FS)$.

B.3. Adaptive prediction sets

We implement adaptive prediction sets following the method in Angelopoulos et al. (2020) with their regularization parameter λ set to 0. To calibrate our *conformal flipper*, we first draw a subset of the validation set (the *calibration set*), apply a random octahedral flip $Q \sim U(\mathcal{O})$ to each calibration shape, and then pass each flipped shape QS through the trained flipper to obtain class probabilities $p_\phi(QS) \in \Delta^2 3$. The *calibration score* for a shape S is the sum of the model’s class probabilities $p(QS)$, ranked in descending order, up to and including the true class i^* corresponding to the ground truth flip Q . We fix a confidence level $1 - \alpha$ and return the $(1 - \alpha)$ -th quantile τ of the calibration scores for each shape in the calibration set. In general, smaller values of $1 - \alpha$ lead to smaller values of τ , which ultimately results in smaller prediction sets at inference time, whereas large values of $1 - \alpha$ lead to larger prediction sets at inference time but with stronger guarantees that these sets include the true flip.

At inference time, we first obtain the flipper model’s output probabilities $p_\phi(S) \in \Delta^{|\hat{\mathcal{R}}|-1}$ for some shape S , then sort $p_\phi(S)$ in descending order and add elements of $\hat{\mathcal{R}}$ to the prediction set until their total mass in $p_\phi(S)$ reaches τ . Intuitively, these sets will be small when the flipper is confident in its prediction and assigns large mass to the highest-probability classes. Conversely, the sets will be large when the flipper is uncertain and assigns similar mass to most classes.

B.4. Motivation for test-time augmentation.

In practice, our trained orienter exhibits discontinuities at certain rotations. To demonstrate this, we perform an experiment on a bench shape, which is symmetric under a 180 degree rotation about the y-axis. We rotate the bench around this axis in increments of 1 degree and pass the resulting shape into our orienter for each rotation. We track two metrics throughout this process:

- The quotient loss of our orienter’s prediction (i.e. the loss function from Problem 3, where we quotient the L_2 loss by the octahedral group). This measures the accuracy of our orienter’s predictions up to an octahedral symmetry.
- The chamfer distance between the shapes obtained by applying the orienter’s output to the rotated bench at subsequent angles of rotation. This detects discontinuities in the orienter’s output.

We plot both of these metrics across the rotation angle in Figure 9. Our orienter exhibits several sharp discontinuities, manifested as spikes in the chamfer distance plot. These discontinuities are associated with spikes in the quotient loss, which are occasionally large. However, these spikes in the quotient loss are highly localized, and our orienter performs well for the vast majority of rotations. These localized spikes motivate our use of TTA to improve our pipeline’s performance. We provide implementation details for TTA in Appendices B.1 and B.2.

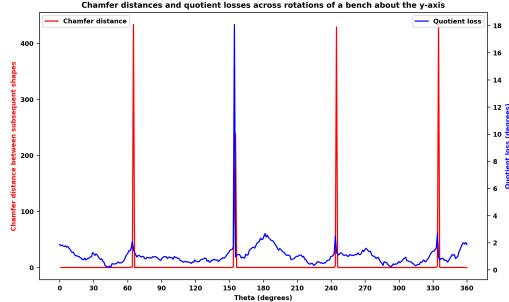


Figure 9: Our orienter exhibits several sharp discontinuities when rotating an input shape about a fixed axis (red curve). These discontinuities are associated with localized spikes in the quotient loss (blue curve), which motivate our use of TTA to improve our pipeline’s performance.

C. Additional figures

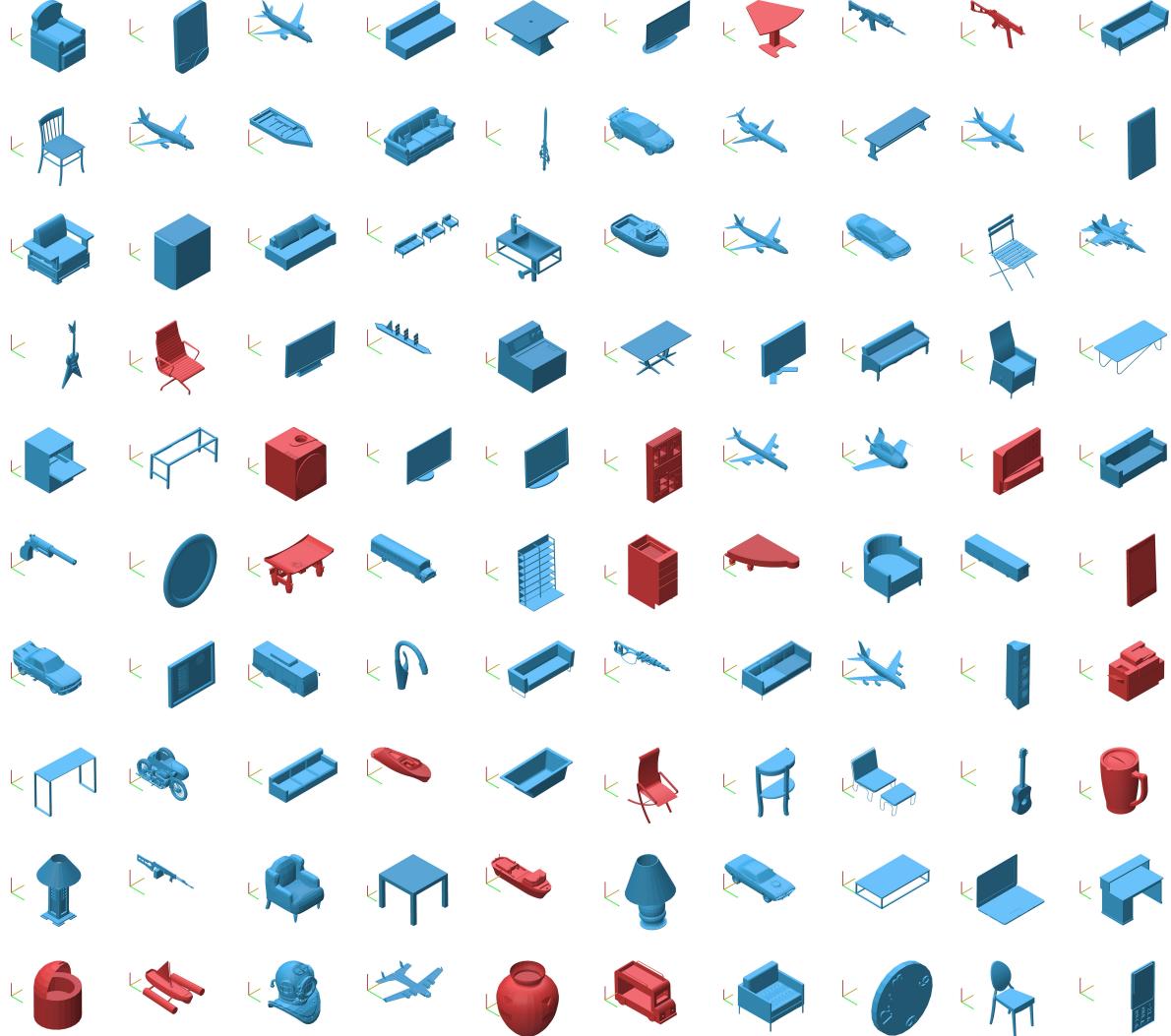


Figure 10: A grid of our model’s outputs given randomly rotated, non-cherrypicked meshes from Shapenet. We render each mesh whose chamfer distance with respect to the ground truth is below a predetermined success threshold in blue, and we render the remainder in red. Our success threshold is the average chamfer distance between a shape randomly rotated by 10° and the original shape; we compute this average across 100 Shapenet validation meshes. Even when predicted shapes do not agree with the ground truth orientation determined by the creators of Shapenet, their orientation is often acceptable in practice.

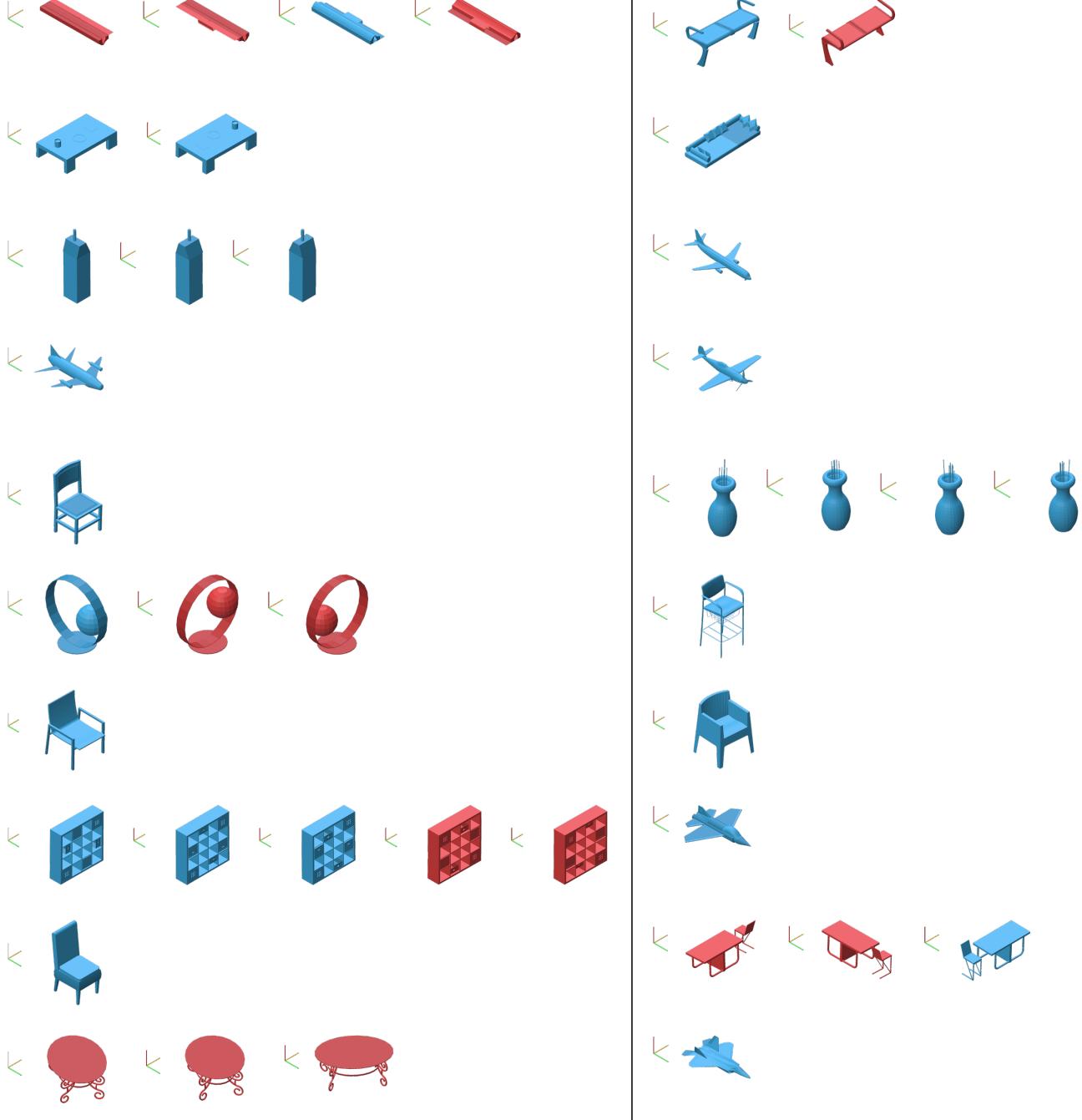


Figure 11: A grid of 20 adaptive prediction sets generated using our pipeline given randomly-rotated, non-cherrypicked meshes from Shapenet. We render each mesh whose chamfer distance with respect to the ground truth is below a predetermined success threshold in blue, and we render the remainder in red. Our success threshold is the average chamfer distance between a shape randomly rotated by 10° and the original shape; we compute this average across 100 Shapenet validation meshes. Our adaptive prediction sets are typically small and include reasonable orientations, which an end user may choose between as needed for their target application.

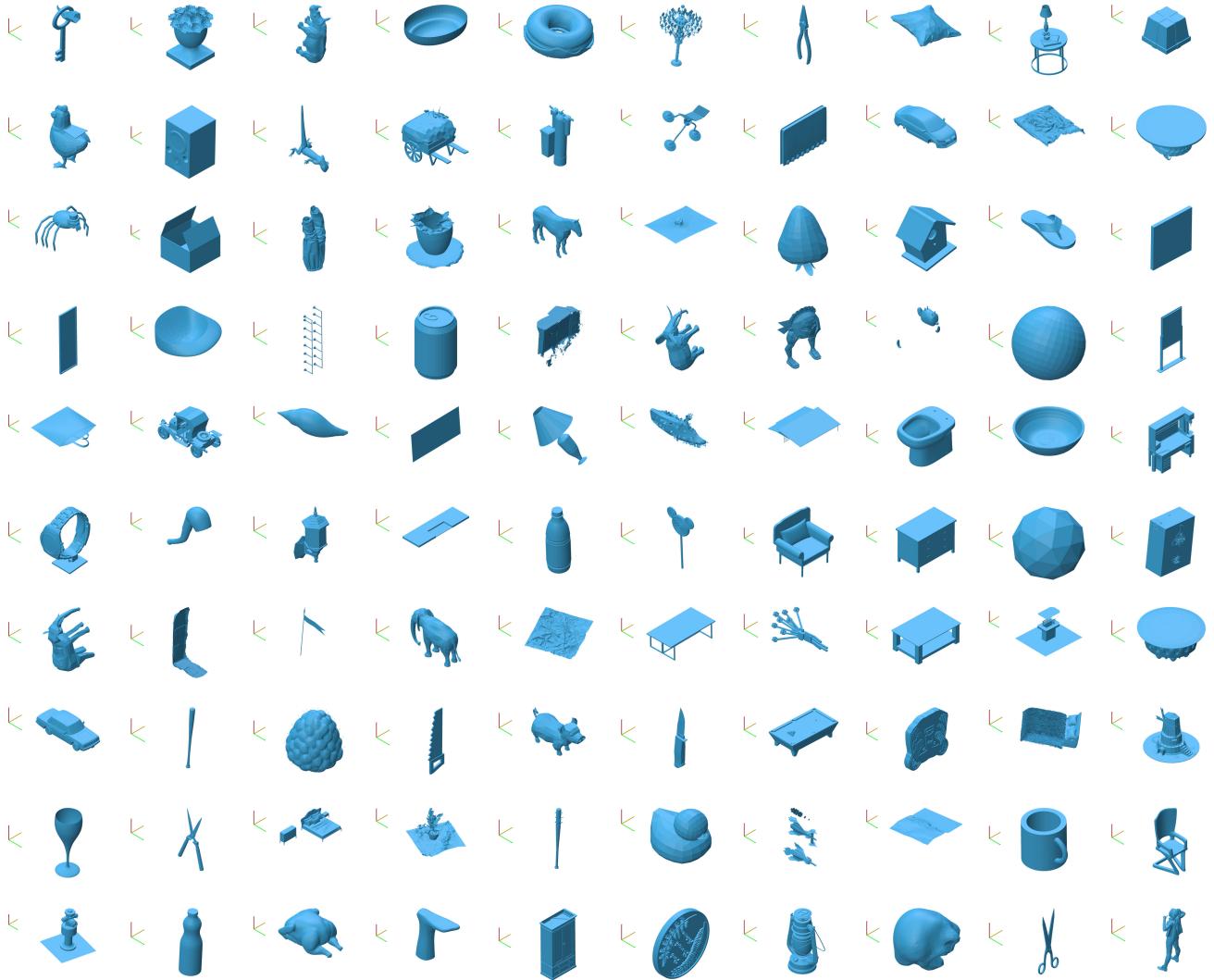


Figure 12: A grid of our model’s outputs given randomly rotated, non-cherrypicked meshes from Objaverse. Our quotient regressor consistently succeeds on out of distribution meshes, as most of our pipeline’s outputs are correctly oriented up to a cube flip. Our flipper has greater difficulty generalizing, but predicts an acceptable flip in many cases. We expect that training on a larger and more diverse dataset of oriented shapes will improve our flipper’s generalization performance.