A Scalable Active Framework for Region Annotation in 3D Shape Collections

Li Yi[1] Vladimir G. Kim[1,2] Duygu Ceylan[2] I-Chao Shen[3] Mengyan Yan[1]
Hao Su[1] Cewu Lu[1] Qixing Huang[4,5] Alla Sheffer[3] Leonidas Guibas[1]
[1]Stanford University [2]Adobe Research [3]University of British Columbia [4]TTI Chicago [5]UT Austin

Figure 1: *We use our method to create detailed per-point labeling of 31963 models in 16 shape categories in ShapeNetCore.*

Abstract

Large repositories of 3D shapes provide valuable input for data-driven analysis and modeling tools. They are especially powerful once annotated with semantic information such as salient regions and functional parts. We propose a novel active learning method capable of enriching *massive* geometric datasets with *accurate* semantic region annotations. Given a shape collection and a user-specified region label our goal is to correctly demarcate the corresponding regions with minimal manual work. Our active framework achieves this goal by cycling between manually annotating the regions, automatically propagating these annotations across the rest of the shapes, manually verifying both human and automatic annotations, and learning from the verification results to improve the automatic propagation algorithm. We use a unified utility function that explicitly models the time cost of human input across all steps of our method. This allows us to jointly optimize for the set of models to annotate and for the set of models to verify based on the predicted impact of these actions on the human efficiency. We demonstrate that incorporating verification of all produced labelings within this unified objective improves both accuracy and efficiency of the active learning procedure. We automatically propagate human labels across a dynamic shape network using a conditional random field (CRF) framework, taking advantage of global shape-to-shape similarities, local feature similarities, and point-to-point correspondences. By combining these diverse cues we achieve higher accuracy than existing alternatives. We validate our framework on existing benchmarks demonstrating it to be significantly more efficient at using human input compared to previous techniques. We further validate its efficiency and robustness by annotating a massive shape dataset, labeling over 93,000 shape parts, across multiple model classes, and providing a labeled part collection more than *one order of magnitude* larger than existing ones.

Keywords: shape analysis, active learning

Concepts: •**Computing methodologies** → **Active learning settings; Shape analysis;**

1 Introduction

In recent years, large and growing online repositories of 3D shapes have proved to be a rich resource for data-driven techniques in computer graphics, vision, and robotics. Recent 3D modeling interfaces, e.g. [Chaudhuri et al. 2011], and advances in text and image understanding [Lin et al. 2014] demonstrate that data-driven tools become especially effective when they rely on big, curated datasets with detailed semantic annotations. However, existing annotated datasets for 3D shapes [Chen et al. 2009] contain only a few of hundreds of models and are also largely composed of manifold, watertight meshes that are not representative of the data available in public repositories such as 3D Warehouse or ShapeNet [Chang et al. 2015]. This significantly limits their practical applications in real-world scenarios. Filling this void requires an efficient tool for *reliably* annotating massive, diverse and growing datasets of polygon soup 3D models. Automatic annotation can never be perfectly reliable as such annotation requires knowledge of detailed shape semantics, yet the size and constant evolution of 3D model databases renders fully manual annotation impractical. We view accuracy as the most critical metric, as all subsequent processing breaks down with inaccurate solutions. Consequently we provide a middle-ground approach: an active-learning method that combines human annotation, algorithmic annotation propagation, and human *verification* of *every* generated annotation.

We focus on generating semantic per-point region labels, such as object parts and salient regions, since they have proved to be extremely useful for various downstream applications. Given a collection of 3D shapes and a user-prescribed label of interest, our system produces human-verified per-point labels for each shape, indicating which points belong to the region of interest (see Figure 1).

While one can clearly add verification as a post-process for any active learning method, such naive addition would lead to a significant increase in human work. Instead, we integrate verification into the active learning framework, by using verification output as part of our feedback loop. This integration ensures accurate annotations, allows to identify human errors, andresults in substantial time savings compared to an equivalent verification-less approach (see Figure 2a and 2b). The number of human annotations required to reliably label the input data goes down dramatically, replaced by a

SA '16 Technical Papers,, December 05-08, 2016, , Macao
ISBN: 978-1-4503-4514-9/16/12
DOI: http://dx.doi.org/10.1145/2980179.2980238

already learned shapes.

← Active learning approach for learning.

the novelty.

combination of automatic annotation and positive verification. This substitution leads to x100 time saving per each avoided annotation without any loss in quality, as substantially less time is required by a human to verify a region than to annotate it.

Our method alternates between four main steps: annotation, propagation, verification, and learning. It obtains manual region annotations for a selected set of models; then automatically propagates these annotations across the input shape database; acquires human verification of both human and automatic annotations for selected models; and finally uses the verification result to improve the automatic propagation algorithm.

When choosing the shapes to present to human workers for annotation and verification we seek to maximize *annotation efficiency*. We define it as the ratio between the number of annotations verified to be correct and the total time spent by human workers. Different from existing work on semi-supervised shape annotation [Wang et al. 2012], we explicitly model the cost of human input, i.e. semantic region demarcation and verification. We use this model to define the annotation efficiency as a utility function, and repeatedly optimize its expected value as we select the next batch of shapes for manual annotation or labeling verification. Our studies show that human verification is as accurate as human annotation (see Section 9), so to detect both human errors and ambiguous models we verify both automatic and manual annotations.

Our algorithm propagates labels from the annotated shapes to unlabeled ones by exploiting both local geometric features and global shape structure. We combine feature-based classifiers, point-to-point correspondences, and shape-to-shape similarities into a single CRF optimization over the network of shapes. Previous techniques that only rely on point-wise shape features [Wu et al. 2014] are sensitive to outliers and large shape variations often observed in massive datasets. By taking advantage of higher-level structural similarities between shapes in addition to low-level point-wise geometric cues, our method achieves superior performance on large and heterogenous datasets (see Figure 2a and 2c).

Our learning step leverages the verified annotation results to update our shape network and improve the performance of the propagation technique in subsequent iterations. In particular we learn a better shape-to-shape similarity metric, and learn better weights for CRF terms, enabling our network of shapes to adapt dynamically to the analyzed category and the label of interest (see Figure 13).

We compare our method to existing alternatives and demonstrate that it can reach the same accuracy with half the human input and scales to much bigger datasets. We test our annotation tool on ShapeNetCore [Chang et al. 2015], a massive and challenging dataset obtained from online repositories. We annotate 16 different shape categories containing 31963 shapes, and produce 93625 verified annotations with only an estimated 110 hours worth of crowdsourced work, which would have taken 780 hours without our tool.

Contribution. Our key contribution is the development of a novel scalable method for efficient and accurate geometric annotation of massive 3D shape collections. Our approach explicitly models and minimizes the human cost of the annotation effort. This contribution is made possible due to two main technical innovations. First, by embedding the verification step as part of our unified throughput maximization framework, we ensure result accuracy and dramatically improve annotation speed. Second we develop and exploit a versatile annotation propagation algorithm that leverages both global inter-shape and local point-wise features over a dynamic shape network, evolving under user feedback. Jointly these contributions significantly reduce the cost of labeling, and allow collecting verified annotations on a massive dataset that is 30 times larger than existing curated 3D shape datasets.

ways they achieve this

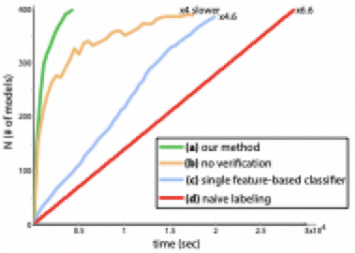

Figure 2: *This figure illustrates the number of correctly-labeled models (y-axis) as people spend more time providing input (x-axis) for a representative collection. Our result corresponds to the highest-performing curve (a), and we provide two variants of our method, one that does not include verification (b), and one that only uses local geometric features to train a single classifier for the entire dataset (c). We also show baseline cost of manually labeling every model (d). Note that all variants take substantially longer to annotate the entire dataset.*

2 Previous Work

Our work builds on previous techniques for analyzing collections of 3D shapes, as well as methods that utilize active learning to address problems such as object detection and classification.

Semantic shape labels. Many existing methods in geometry processing require semantic shape labels as part of their input. For example, libraries of 3D models with annotated semantic parts have been used for modeling-by-example [Chaudhuri et al. 2011], to synthesize shape variations [Kalogerakis et al. 2012], and to complete geometry from partial scans [Sung et al. 2015]. These methods require accurate part annotations, and currently rely on manually annotated databases. Existing work on non-expert annotation of 3D models using a crowdsourcing interface reports processing times of 3 minutes per worker to segment a shape into unlabeled parts [Chen et al. 2009], similarly, our region annotation interface requires 30 seconds per worker per region label (see Section 9). These timings render full manual annotation of large and growing collections impractical. Our active learning approach that incorporates verification allows generating massive collections of labeled region data with the same accuracy as full manual annotation (see Table 3).

Unsupervised shape segmentation. Previous work on unsupervised shape analysis has largely focused on segmenting models into compatible and geometrically separable parts. While earlier approaches segmented individual shapes [Shamir 2008], several recent works have shown that joint analysis of a group of related objects provides better geometric cues for what constitutes a semantically meaningful part [Sidi et al. 2011; Huang et al. 2011; Hu et al. 2012; Huang et al. 2014; Shu et al. 2016]. These methods typically do not associate the resulting segments with semantic labels, making their as-is outputs less useful for semantics driven applications. In contrast, we seek to identify semantic regions, which may contain multiple parts, or a fraction of a part, and may have boundaries that do not align with sharp geometric features. More importantly, it is typically not possible to generate perfectly accurate segmentations automatically, whereas our goal is to produce accurate annotations by including a human verification step.

Supervised shape analysis. Supervised learning is frequently used to obtain semantic annotations of 3D shapes [Kalogerakis et al. 2010; Lv et al. 2012; Xie et al. 2014; Makadia and Yumer 2014; Guo et al. 2015]. These methods employ traditional machine learn-

ing techniques and construct classifiers based on geometric features. Since any classifier can generate erroneous labels we provide an efficient technique that allows humans to verify every output. The performance of supervised techniques greatly depends on the availability of annotated training data that covers all shape variations. Currently, such datasets are assembled manually and are limited in scale. We develop an efficient tool to reliably annotate massive and diverse datasets from scratch, providing valuable data for future supervised analysis research.

Active and interactive image analysis. Active methods have been widely explored in image analysis. Their main advantage is that they acquire training data incrementally, enabling a classifier to predict which samples need to be labeled next to improve their performance [Vijayanarasimhan and Grauman 2008; Branson et al. 2011; Vezhnevets et al. 2012; Branson et al. 2014]. While these methods focus on improving classifiers, we focus on the end-goal of generating accurate, verified results, within a fixed human-effort budget. To achieve this goal, unlike previous active learning techniques we explicitly model human behavior in our annotation and verification phases through an integrated objective.

Recently, Russakovsky et al. [2015] proposed using a mixture of verification and refinement tasks for manually fixing failures of automatic image tagging algorithms. They use human input to detect and correct errors, but do not take advantage of this input to improve automatic labeling. In contrast, our approach utilizes both types of human input to improve label propagation. Our ability to harness verification output within an active learning framework is particularly useful for geometry processing where human verification is two orders of magnitude faster than annotation and where each annotation takes 30s to complete (see Section 9). For comparison, for images Russakovsky et al. [2015] cite a ratio of two between these tasks (10s for annotation and 5s for verification).

Active learning for shape analysis. There are very few methods that leverage active learning to annotate or classify 3D models, and, as pointed out in these works, 3D raises very different challenges from images and other 2D data. Wang et al. [2012] use active learning to segment shapes into labeled parts. They iteratively ask users to select pairs of patches and indicate whether they have the same label to actively co-segment 3D shapes. In contrast to their framework where many models are segmented with no direct human input we verify each result, confirming accuracy. We further show that even when using verification in-the-loop our method can annotate same-scale datasets using about half the human input time (see Section 9 for detailed comparison). Wu et al. [2014] offer a simpler painting interface to query the users, using a similar setup and yielding similar accuracy and efficiency to Wang et al. Boyko et al. [2014] use group verification to speed up object classification in urban point clouds. Their approach is not applicable in our setting, since their input is pre-segmented into objects, and they focus on classifying these segments.

Correspondence networks. Our automatic label propagation technique builds on recent advances in correspondence networks that enable establishing semantic relations in collections of 3D models [Kim et al. 2012] and images [Rubinstein et al. 2012] via pairwise matching and various notions of cycle consistency [Huang et al. 2014]. These works focus on designing static networks that remain fixed throughout the subsequent processing. In contrast, we develop and employ a dynamic network that changes as we acquire more human input, gradually improving our shape-to-shape similarity and relative weighting of point-wise shape features and global correspondences. At the end we learn a network that works the best for the label of interest on the input shape collection.

3 Active Learning Framework Overview

Our active learning framework produces human-verified per-point annotations of collections of 3D shapes while minimizing user supervision. The input to our system is a set of shapes in the same category (e.g., chairs) and a label of interest (e.g., "back"), and the output is a per-point boolean value that indicates whether the point belongs to the label of interest. To generate these labels for polygon soup inputs, we uniformly sample 3000 points on each shape and project per-point labels to per-face labels in a post-processing step.

Our approach reduces annotation effort by automatically propagating acquired annotations to unlabeled shapes. To achieve accurate results all labels are further verified by a human. For each automatically computed labeling that is verified to be correct we avoid an expensive human annotation, replacing it by a cheaper verification task. We use human input to improve the automatic propagation, and use the propagation algorithm to guide the decision of what human input to obtain next. Our approach iteratively alternates between these two steps, until all models are positively verified or the human-work budget is spent.

At each iteration we address four critical problems: issuing most budget-effective human tasks, quickly obtaining human input, effectively propagating the input to unlabeled shapes, and leveraging automatic propagation results and human input to improve the propagation in the next iteration. Below we provide an overview of the methods that we use to address each of these problems; Figure 3 depicts our pipeline.

Our system optimizes for two types of human tasks, annotation and verification, with the goal of maximizing framework efficiency, measured as a ratio between verified correct annotations and invested human time. To enable crowd-sourcing we execute the human tasks in batches, and thus at each iteration m we separately optimize for an annotation set $\mathcal{A}^m$ and a verification set $\mathcal{V}^m$. Since we do not have a-priori knowledge of which annotations will be supplied or verified as correct, we model the propagation of annotated or verified labels using a probabilistic formulation, and maximize the expected efficiency of the framework. This approach enables us to unify both tasks within a single utility function, which we optimize twice in our pipeline: first to decide which shapes to annotate for and second, after annotations are collected and automatically-propagated, to decide which shapes to verify (see Section 4).

To obtain human input we devise a web-based interface for each task. We designed our interface to be suitable for non-expert users and issued simple tasks to facilitate crowd-sourcing. For the annotation task, our interface presents workers with shapes in the annotation set $\mathcal{A}^m$ one-by-one and asks the user to highlight the part of interest, producing per-point labels h^m. For verification tasks we expect the majority of models in verification set $\mathcal{V}^m$ to have correct labels, and thus, to save worker time, we show viewers the entire set of models to verify at once and only ask them to select the incorrectly labeled ones. For each shape in the verification set our interface produces a value indicating whether its prediction is correct Q^m. See supplemental video and Section 5 for details.

We propagate human annotations to unlabeled shapes over a similarity network between shapes by leveraging multiple cues. First, we train classifiers for each human annotated shape that utilize local geometric feature similarities. Second, we incorporate point-to-point correspondences estimated among shapes connected in the shape network to provide global structural and contextual information. Finally, we leverage smoothness priors to favor regions of interest that are compact and continuous. We learn which of these cues are more trustworthy for a particular label as we acquire more human input. This novel combination is made possible with an

Over say what they are doing
polygon soup meaning.
ratio of verified correctness Vs human time.
first annotate, then verify
supplemental video
Correspondence networks

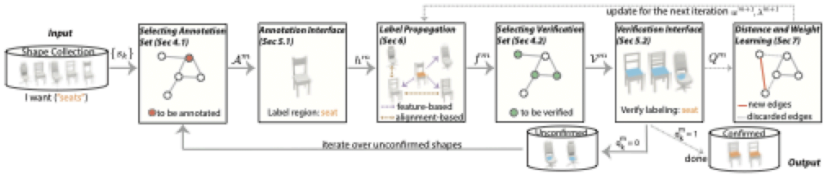

Figure 3: *This figure summarizes our pipeline. Given the input dataset we select annotation set and use our UI to obtain human labels. We automatically propagate these labels to the rest of the shapes and then query the users to verify most confident propagations. We then use these verifications to improve our propagation technique.*

efficient CRF formulation where the nodes are the shape points. Our CRF data term is defined based on the output of the feature-based classifier trained on the most similar annotated shape, and the pairwise term is derived from (i) point-to-point correspondences weighted by network-based shape similarity and (ii) spatial proximity. Our propagation procedure leverages per-point human labels h^m obtained via the annotation interface to predict labels for all points f^m that have no annotations. The setup and the optimization are described in details in Section 6.

We dynamically adapt our shape network depending on type of label that is being propagated and shape category. In particular, after each verification stage, shape similarities and the network are updated to reflect this human input. We similarly learn the relative weighting of the different terms in our CRF to ensure that it adapts well to new data. We start with empirically-chosen initial parameter values and then update them at every iteration (see Section 7).

Pipeline. We combine these four fundamental modules into a joint pipeline. At each iteration m we first construct an annotation set $\mathcal{A}^m$ and obtain human input h^m for each shape in the set. We then propagate these annotations to the remaining models and get automatic predictions f^m. We select a subset of these predictions and all unverified human annotations to be verified $\mathcal{V}^m$ and obtain human verifications for those predictions Q^m ($q_k^m = 1$ if shape s_k is positively verified by the current iteration m). Finally, we compare automatic predictions f^m and human verifications Q^m to update shape-to-shape similarities $\omega_{i,j}^{m+1}$ and CRF weights λ^{m+1}. Once a model is positively-verified, we do not select it for subsequent annotation or verification. If a shape was manually annotated, but not verified, we mark it as ambiguous and exclude from analysis. Our pipeline terminates once all models are positively verified.

4 Selecting Annotation and Verification Sets

The goal of our active learning optimization is to select shapes for *annotation* and *verification* to maximize the human work efficiency at each iteration. We measure this efficiency as the ratio between positively verified shapes N_{good}^m and time investment T^m at the end of the iteration. Our challenge is that N_{good}^m is a function of verification result Q^m, which cannot be obtained before we compute the optimal human tasks and collect human annotations. To address this challenge we treat human input Q^m as random variables that depend on annotation and verification sets $\mathcal{A}^m, \mathcal{V}^m$. Hence both N_{good}^m and T^m are also random variables with respect to $\mathcal{A}^m, \mathcal{V}^m$, and we define our utility function based on their expected values:

$$E_U^m(\mathcal{A}^m, \mathcal{V}^m) = \frac{\mathbb{E}[N_{\text{good}}^m | \mathcal{A}^m, \mathcal{V}^m]}{\mathbb{E}[T^m | \mathcal{A}^m, \mathcal{V}^m]}. \quad (1)$$

We observe that all variables at the previous iteration $m-1$ are known at iteration m, and thus we only need to compute the expected value of the change from the previous iteration:

$$\mathbb{E}[N_{\text{good}}^m | \mathcal{A}^m, \mathcal{V}^m] = N_{\text{good}}^{m-1} + \sum_k (1 - q_k^{m-1}) \mathbb{E}[q_k^m | \mathcal{A}^m, \mathcal{V}^m].$$

Here we write $Q^m = \{q_k^m\}$, where $q_k^m = 1$ if the shape s_k is verified as correct by iteration m and 0 otherwise. The summation term adds the expectation over whether a shape will be verified $\mathbb{E}[q_k^m | \mathcal{A}^m, \mathcal{V}^m]$ for each previously unverified shape.

We compute the change in total time spent by humans as:

$$\mathbb{E}[T^m | \mathcal{A}^m, \mathcal{V}^m] = T^{m-1} + \tau_{\text{ann}} |\mathcal{A}^m| + \sum_k t_{\text{ver}}(\mathbb{E}[q_k^m | \mathcal{A}^m, \mathcal{V}^m]) v_k^m.$$

We define the verification set V^m as $V^m = \{v_k^m\}$, where $v_k^m = 1$ if the shape s_k is selected to be verified in iteration m. We use $\tau_{\text{ann}} = 30$s as a constant time to annotate a shape, as estimated via our user study. For every verified shape, we model the cost of verification by observing that a person can quickly skim through the list of correct results, but would need to spend extra effort to click on an incorrect one:

$$t_{\text{ver}}(q_k^m) = \tau_{\text{ident}} + (1 - q_k^m)\tau_{\text{click}}, \quad (2)$$

We use $\tau_{\text{ident}} = 0.3$s as the time required to identify whether the annotation is correct (as the crowd worker scans through a sequence of results) and $\tau_{\text{click}} = 1.1$s denoting the time required to select the incorrectly annotated shape by a click action. Both constants are average values estimated via user studies.

Our remaining challenge for evaluating the utility function in Equation 1 is to compute the expected verification $\mathbb{E}[Q^m | \mathcal{A}^m, \mathcal{V}^m]$. We approximate this value based on the confidence of our automatic propagation procedure $C^m[k]$ defined for every shape s_k. Note that estimating this confidence when selecting the annotation set is challenging since we have not yet executed the propagation algorithm but a more reliable value can be computed when we optimize for verification set. Thus, we use two different approximations for our confidence value, pre-propagation confidence C_{ann}^m when we optimize for the annotation set, and post-propagation confidence C_{ver}^m when we optimize for the verification set. We then use isotonic regression [Zadrozny and Elkan 2002], denoted by function $r(\cdot)$ to calibrate the confidences and obtain expected values: $\mathbb{E}[Q^m | \mathcal{A}^m, \mathcal{V}^m] = r(C^m)$. We utilize positively-verified models as training data for this regression using F1 scores (harmonic mean of precision and recall) as true prediction confidences.

Next, we provide details for estimating confidences and selecting the annotation and verification sets (see Figure 4 for examples).

4.1 Selecting the Annotation Set

We optimize for an annotation set that maximizes the utility function: $\mathcal{A}^m = \text{argmax}_{\mathcal{A}} E_U^m(\mathcal{A}, \mathcal{V}^m)$.

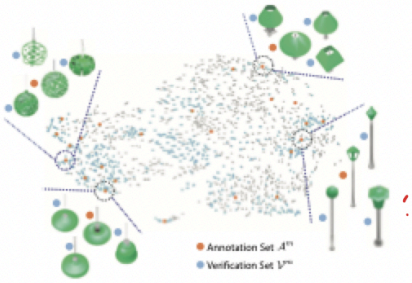

Figure 4: *At each iteration our method selects an annotation set (in orange) and a verification set (in blue) from the shape network (in gray). The models selected for annotation are distributed over the network to provide good coverage of shape variations. In contrast, models selected for verification tend to cluster close to annotated models since labels can be more reliably propagated between them.*

Pre-propagation confidence. In order to optimize our utility function we need to compute the expected verification outcome for each shape. This value depends on how confidently the automatic predictions are made for each shape. Our method leverages two terms to propagate annotations between shapes: (i) feature-based propagations from the most similar annotated shape, and (ii) point-to-point correspondences weighted by global shape similarity. Intuitively, in our first term the confidence of the predictions, c_k, for each shape s_k is likely to increase as the similarity, $\omega_{k,x}$, between this shape, s_k, and its most similar annotated shape, s_x, increases. The second term, on the other hand, is more global, enforcing similar predictions, and thus similar prediction confidences, for shapes that share correspondences. Combining these two intuitions, we estimate, $C^m_{\text{nn}} = \{c_k\}$, the prediction confidence for each shape as:

$$C^m_{\text{nn}} = \arg\min_c \sum_k \| \max_{x \in \mathcal{A}^m} \omega^m_{k,x} - c_k \|^2_2 + \lambda^m_1 \sum_{k,j} \omega^m_{k,j} \| c_k - c_j \|^2_2,$$

We weight the second term by λ_1, the same weight that is used for correspondence term in the CRF optimization. Figure 5a demonstrates the correlation between our estimate C^m_{nn} and true q^m values obtained from ground truth at iteration $m = 0$ for a representative dataset. Please note that no human input was collected at this stage to produce this confidence estimation.

Given any candidate annotation set, we can use this confidence estimate to compute the optimal verification set as described in Section 4.2 and compute the utility. Annotation set optimization thus can be performed by generating multiple candidate sets and choosing the one with maximum utility, as described next.

Annotation set optimization. Our goal is to select an annotation set $\mathcal{A}^m$ that will maximize the utility function, i.e. we choose $\mathcal{A}^m$ to consist of models that are expected to *confidently annotate* their unlabeled neighbors. We select a fixed number of shapes at each iteration $|\mathcal{A}^m| = n_{\text{ann}}$ (where $n_{\text{ann}} = 0.01N$). To perform such a selection we use a variant of beam search, where at any time we keep a set of sub-optimal solutions represented by $N = |\{s_k\}|$ candidate subsets Ω_k. Initially, the subsets just include one shape ($\Omega^0_k = \{s_k\}$). At every iteration $i > 0$, we evaluate every possible k^{th} subset that includes the shape s_k and any set Ω^{i-1}_l from the

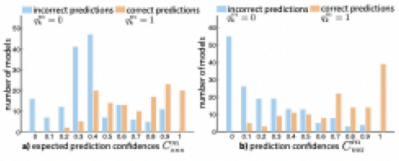

Figure 5: *Given a dataset of models with ground-truth human verifications Q^m, we demonstrate the distribution of our prediction confidences, C^m (x-axis) (blue bars for incorrectly propagated predictions and brown for correct). A good confidence predictor is expected to place higher blue bars closer to 0 on the x-axis, correctly identifying the bad predictions, and higher yellow bars closer to 1 on the x-axis, correctly identifying the good predictions. (a) While the estimates used to select the annotation set in the beginning, i.e. with no human input yet, are less accurate, they are sufficient to bootstrap our algorithm. (b) These estimates significantly improve after a single iteration of annotation propagation enabling to optimize for a more reliable verification set.*

previous iteration, and pick the one that maximizes utility function:

$$\Omega^i_k = \underset{\mathcal{A}^m = \Omega^{i-1}_l \cup \{s_k\}, l=1\ldots N}{\text{argmax}} \mathbb{E}_U(\mathcal{A}, \mathcal{V}),$$

where $\mathbb{E}_U(\mathcal{A}, \mathcal{V})$ is defined by Equation 1. When the cardinality of Ω^i_k reaches n_{ann} we greedily pick a shape to remove (maximizing the utility function for a subset with cardinality $n_{\text{ann}} - 1$), thus at next iteration when we add a new element our cardinality remains constant $|\Omega^{i \geq n_{\text{ann}}}_k| = n_{\text{ann}}$. This element removal allows us to refine our subsets and we terminate either once subsets Ω stop changing or after $10 n_{\text{ann}}$ iterations. We set $\mathcal{A}^m$ to be the subset that maximizes the utility function.

4.2 Selecting the Verification Set

After we obtain user input for our annotation set and propagate the labels, we compute our verification set by maximizing the same utility function: $\mathcal{V}^m = \text{argmax}_{\mathcal{V}} E^m_U(\mathcal{A}^m, \mathcal{V})$; this time using $\mathcal{A}^m$ as a fixed constant.

Post-propagation confidence. We now refine our confidences by leveraging the results of label propagation. In particular, we use the per-point label probability $\rho_p = \Pr(f^m(p) = 1 | h^m)$ computed by our propagation step (see Section 6). We denote prediction entropy: $J(p) = \rho_p \log \rho_p + (1 - \rho_p) \log(1 - \rho_p)$, a value that increases for more confident predictions as ρ_p gets closer to 0 or 1, and we further normalize it to be a positive value in a range $[0, 1]$: $\bar{J}(p) = 1 + \frac{J(p)}{\log(2)}$. We now use this normalized entropy to get refined confidences:

$$C^m_{\text{val}}[k] = \frac{1}{|s_k|} \sum_{p \in s_k} \bar{J}(p) \cdot C^m_{\text{nn}}[k],$$

where $|s_k|$ is the number of points sampled on a shape s_k. Note that the correlation between the ground truth confidences and our estimations improves when optimizing for the verification set since we have more input obtained from the annotation propagation step (see Figure 5). We use these confidence estimates when optimizing for the verification set.

Verification set optimization. Since our approach directly benefits from positive verification and does not benefit from a negative one, we select a verification set $\mathcal{V}^m$ to contain annotations that are most likely to be positively-verified. We consequently initialize the