HiT: Hierarchical Transformers for Unsupervised 3D Shape Abstraction

Aditya Vora[1] Lily Goli[2] Andrea Tagliasacchi[1,2] Hao Zhang[1]

[1]Simon Fraser University [2]University of Toronto

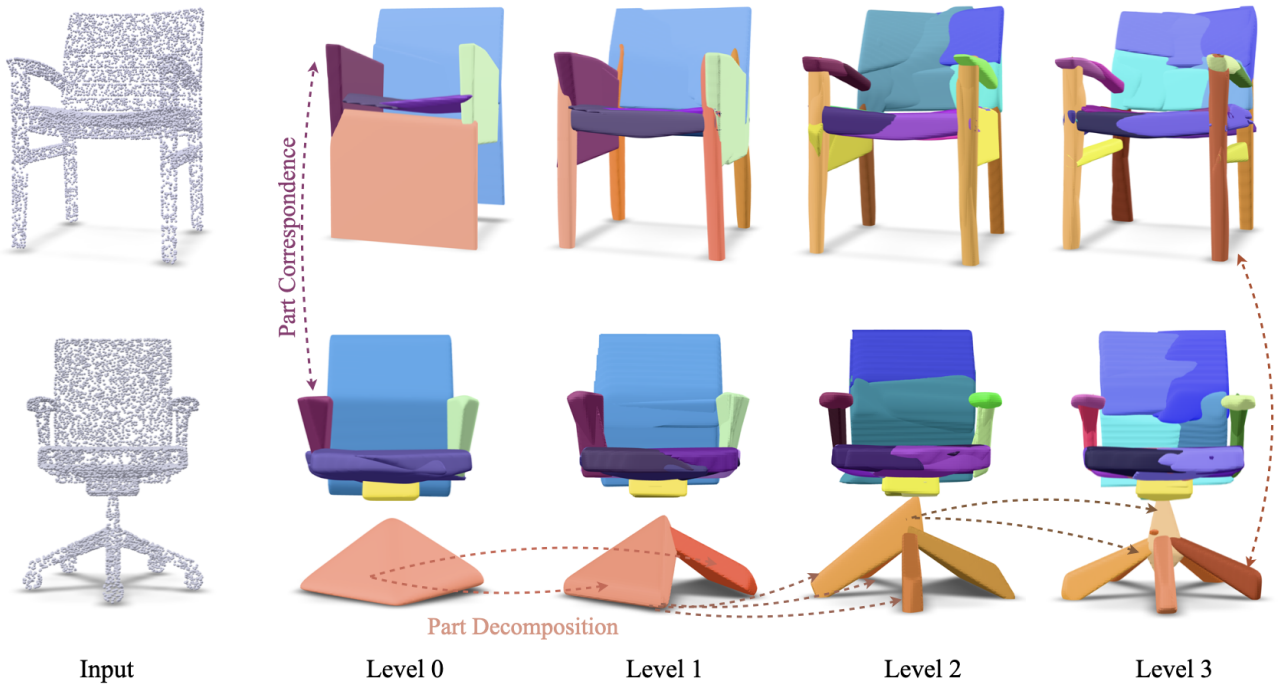


Figure 1. We present an attention-based architecture for hierarchical part abstraction of 3D objects. Our model flexibly adapts the number of parts, without supervision, allowing semantically similar regions (e.g. chair legs) to be decomposed, through the hierarchies, into different numbers of child parts depending on their geometry. While the resulting parts at finer levels of abstractions may no longer be semantic (e.g., split of the chair seats), part correspondences between shapes remain meaningful at different levels of the hierarchies.

Abstract

We introduce HiT*, a novel hierarchical neural field representation for 3D shapes that learns general hierarchies in a coarse-to-fine manner across different shape categories in an unsupervised setting. Our key contribution is a hierarchical transformer (*HiT*), where each level learns parent–child relationships of the tree hierarchy using a compressed codebook. This codebook enables the network to automatically identify common substructures across potentially diverse shape categories. Unlike previous works that constrain the task to a fixed hierarchical structure (e.g., binary), we impose no such restriction, except for limiting the total number of nodes at each tree level. This flexibility allows our method to infer the hierarchical structure directly from data, over multiple shape categories, and representing more general and complex hierarchies than prior approaches. When trained at scale with a reconstruction loss, our model captures meaningful containment relationships between parent and child nodes. We demonstrate its effectiveness through an unsupervised shape segmentation task over all 55 ShapeNet categories, where our method successfully segments shapes into multiple levels of granularity.*

Project Page: aditya-vora.github.io/HiT/

1. Introduction

It is well established in cognitive science that humans perceive shapes as structured collections of parts, organized *hierarchically* [15, 28]. This hierarchical part cognition helps to reason about the function of each part [23], and allows humans to effortlessly establish correspondences between similar parts in different shapes in a collection. Introducing similar hierarchical decompositions to 3D digital shapes not only aligns representations with human perception, but also enables key applications in computer graphics and robotics. In graphics, it facilitates part-aware editing [13], attribute transfer [41], and compositional shape generation [3]; in robotics, it can support affordance reasoning, manipulation planning [20], and generalizable interaction with objects [33], in a coarse-to-fine manner.

While several prior works [19, 24, 42] achieve hierarchical part learning through direct supervision, their reliance on labeled 3D datasets limits the scalability and generalizability of these methods. A practical alternative must be an unsupervised and generalizable representation trained across shapes, enabling part correspondences to emerge as in human perception. Current unsupervised methods often frame part representation learning as shape reconstruction task using implicit part representations [5, 9, 12, 27], with priors imposed via constrained part forms – e.g., convex shapes [9] or low-capacity MLPs [5].

While effective for single-level decompositions, multi-level extensions usually assume a fixed, often binary, tree structure [12, 27], which is unnatural and fails to capture the diversity of real-world object hierarchies. For instance, number of chair legs can vary between instances; see figure 1. Capturing this variability requires learning the hierarchy itself, and *not* prescribing it.

Meanwhile, recent transformer-based models such as NeuMap [35] have effectively shown using attention to learn "soft" spatial correspondences in a flexible, data-driven way. These models suggest a promising path toward flexible, learnable structure, but they *lack an explicit notion of hierarchy* or part-based abstraction. This leaves open the question of how to combine the flexibility of attention-based models with the interpretability and structural grounding of hierarchical part reasoning.

We propose HiT, a Hierarchical Transformer that performs *multi-level* part decomposition by not only learning recurring parts across a shape collection at different levels of abstraction, but also *modeling the relationships between parts* at successive levels of abstraction. At inference time, given an input shape in the form of a point cloud, HiT decomposes it into multiple parts at each level and dynamically assigns a tree structure tailored to that shape; see figure 1.

Inspired by recent advances in transformer-based correspondence learning, HiT is built as a multi-layered transformer decoder, where each layer represents a set of parts using a learned codebook. Part–subpart relationships across levels are established via standard cross-attention, which softly assigns each subpart to a parent part at the level above. Each part is then mapped to a 3D convex primitive that provides a geometric description for that part, with subpart primitives encouraged to lie spatially within their assigned parent.

We show that HiT achieves state-of-the-art part decomposition performance on the ShapeNet/PartNet benchmarks, while producing *geometrically interpretable* (as they are simply convex decompositions at each level) hierarchical abstractions of 3D shapes.

2. Related work

Decomposing shapes into parts is a central problem in 3D understanding. We review methods that progress from unsupervised single-level segmentation to structured hierarchical representations, cover primitive-based implicit models for part reconstruction, and transformer-based approaches to hierarchical modeling in other domains. Finally, we review segmentation methods that utilize large pre-trained models.

2.1. Structured neural shape representations

Single level 3D shape structures. A common approach to 3D shape understanding is to reconstruct them as compositions of primitives or semantic parts. Prior works learn such part-aware representations by approximating implicit surfaces with fixed or learned primitives. For example, [11, 12, 29, 36] represent shapes as unions of superquadrics or Gaussians, but their fixed structures often fail to capture surfaces accurately. BAE-Net [5], Neural-Parts [30], and DAE-Net [7] improve flexibility using MLPs to represent parts, either through branching layers or by parameterizing deformations of simple shapes with invertible neural networks as learned homeomorphisms. While more expressive, these methods often yield coarse decompositions and can collapse without good initialization. CvxNet [9] and BSP-Net [6] balance expressiveness and precision by using convex primitives, which are differentiable and support fine-grained decomposition, but only at a single level. Our approach extends this idea by learning a multi-level hierarchy of convex primitives with a transformer. Each level captures part composition and shared substructures, and a learned codebook enables reusable parts across shapes. This hierarchical design allows unsupervised part segmentation to emerge naturally from the representation.

Multi-level hierarchical 3D shape structure. Part-whole hierarchies provide a deeper understanding of shape structure by modeling how fine parts group into larger components. GRASS [19] and PartNet [42] predict binary part trees using supervised recursive neural networks. StructureNet [24] generalizes this to graph-based hierarchies, but still requires

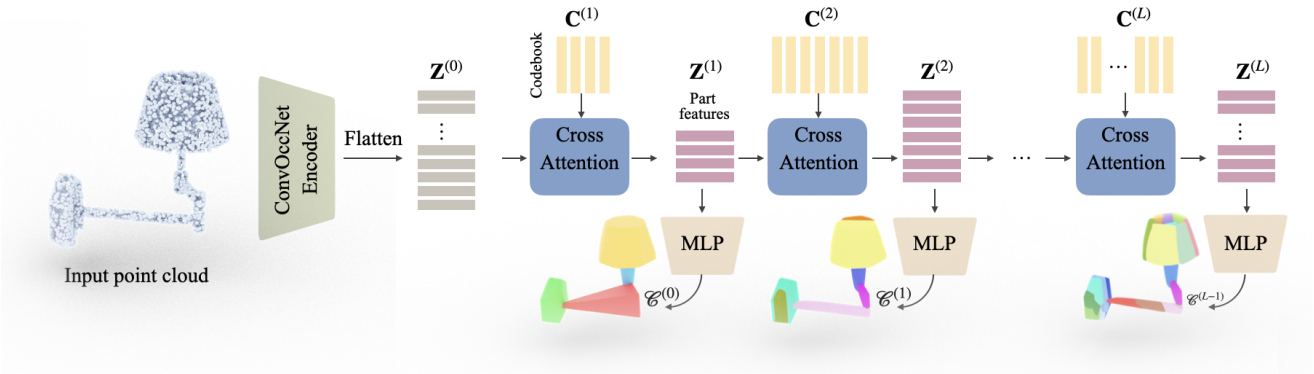


Figure 2. We propose a hierarchical transformer that learns part codebooks at each level, representing shapes from coarse to fine when trained across shapes. Cross-attention "connects" levels, *establishing learnable part–subpart relationships*. The decoded parts are mapped to 3D convex primitives that provide geometric explanations. An example decomposition of a lamp is shown across three levels, from a coarse base and shade to finer structural details.

full supervision of part relationships. RIMNet [27] takes an unsupervised approach to binary decomposition using implicit functions, while [38] explores co-hierarchical analysis of shape sets given pre-segmented parts. In contrast to all these methods, our model learns general n-ary hierarchies, without part annotations, and scales to diverse shape categories, discovering reusable substructures and coarse-to-fine part groupings directly from shape geometry.

Transformers for hierarchical modeling. Transformers have recently been adapted to capture structural and hierarchical information across various domains. GLOM [14] introduced the idea of using a capsule-like transformer for hierarchically modeling part–whole relationships in compositional visual entities. In natural language processing, [26] employed hierarchical transformers for long-range sequence modeling, while in computer vision, models such as Swin Transformer [21] leverage hierarchical attention to capture multiscale spatial dependencies for object representation learning tasks. Building on these lines of work, we propose a hierarchical transformer architecture specifically designed for learning structured representations of 3D shapes. Each level of our transformer captures parent–child relationships between parts through a shared codebook, enabling hierarchical reasoning and compositional generalization across shape categories.

2.2. Shape segmentation with pre-trained models

Several recent methods infer 3D segmentation by importing information external to raw geometry. Multi-view neural-field approaches lift 2D semantic or panoptic masks from posed images into an implicit 3D volume, achieving scene-level labels without part annotations [16, 18, 32, 37, 39, 44]. Complementary work leverages vision-language training to segment shapes under zero-shot or open-vocabulary setups, from point-cloud labeling [4, 17, 22] to CLIP-based mesh highlighting [1, 8, 10]. All of these methods require either 2D mask supervision or large language–vision models, whereas our method learns part hierarchies directly from geometry without any auxiliary cues.

3. Method

Given an input point cloud $\mathcal{X} \in \mathbb{R}^{M\times 3}$, our goal is to recover a *hierarchical* decomposition of $\mathcal{X}$ into disjoint *parts*, across L levels. We represent parts via 3D occupancy fields, and we require the decomposition to be semantically *consistent* across shapes; see figures 3 to 5. At each level ℓ, the shape is abstracted into a set of N_ℓ disjoint parts $\{P_p^{(\ell)}\}_{p=1}^{N_\ell}$. Every parent part $P_p^{(\ell)}$ has a non-empty set of sub-parts $\mathcal{S}_p^{(\ell)}$ at level $\ell+1$, such that,

$$\bigcup_{P_s^{(\ell+1)}\in\mathcal{S}_p^{(\ell)}} \left\{P_s^{(\ell+1)}\right\} \subset P_p^{(\ell)}. \tag{1}$$

Therefore, each parent is decomposed into sub-parts that are fully contained within it. The number of parts N_ℓ is a hyper-parameter specified by the user, and it typically increases with ℓ, enabling progressively finer decompositions. We impose *no further constraints* on the structure of the hierarchy tree beyond the total number of parts at each level. We propose a self-supervised encoder-decoder that discovers hierarchical part decompositions by reconstructing the occupancy field of shapes. The resulting hierarchies emerge without supervision from part labels or predefined tree structures, and naturally adapt to shapes across categories.

We achieve this by introducing Hierarchical Transformer (HiT) architecture, which features a L-layer decoder. Each layer represents one level of the hierarchy using a learnable codebook of N_ℓ parts. To model hierarchical structure, cross-layer attention captures the relationships between parts and their subparts. Each part is then grounded geometrically

by mapping it to a 3D convex primitive, which is required to be fully contained within its parent. The union of all convexes at a given level approximate the full shape. In practice, HiT builds a differentiable tree structure, where each node corresponds to a part embedding, part–subpart relationships are softly encoded in the cross-attention matrix, and each node is grounded by mapping to a localized convex region in 3D space; see figure 1.

Outline. In section 3.1, we describe how decoder layers use part codebooks and cross-attention to define part–subpart relationships. In section 3.2, we show how parts are geometrically grounded using 3D convex primitives with nested containment. Finally, section 3.3 outlines the training objectives that combine reconstruction loss with regularizers to achieve self-supervised training.

3.1. Hierarchical parts transformer

Our hierarchical part transformer consists of a standard point cloud encoder followed by L-layer part hierarchy decoder. The decoder is built around two key decoder components:

(i) codebook-based information bottlenecks at each level, which learn semantically consistent part codes when trained on multi-category shape collections, and

(ii) a cross-attention mechanism that models part–subpart relationships across levels. We now describe each component in detail.

The architecture begins with a point cloud encoder adopted from ConvOccNet [31]. The encoder maps each point to a D-dimensional latent feature, which are then pooled into a voxel grid $\mathbf{G}$ at a fixed resolution R using average pooling within each voxel. The resulting grid is flattened into a feature matrix $\mathbf{Z}^{(0)}$ with dimensions $R^3 \times D$, which serves as the input single-part representation at level zero of the L layers deep HiT decoder hierarchy.

Each decoder level ℓ contains a fixed-size learnable codebook $\mathbf{C}^{(\ell)}$ of code parts that aim to capture recurring shape patterns at that level of abstraction. To achieve this, the N_ℓ codes in the codebook act as queries into the incoming part features $\mathbf{Z}^{(\ell-1)}$ from the previous level, enabling a *soft assignment* of features to the parts represented by their codes:

$$\begin{aligned} \mathbf{Z}^{(\ell)} &= \mathbf{A}^{(\ell)} \cdot \mathbf{V}^{(\ell)}, \\ \mathbf{A}^{(\ell)} &= \mathrm{SoftMax}\left(\frac{\mathbf{Q}^{(\ell)} \cdot \mathbf{K}^{(\ell)\top}}{\sqrt{D}}\right), \\ &\begin{cases} \mathbf{Q}^{(\ell)} = \mathbf{W}_{\mathbf{Q}}^{(\ell)} \cdot \mathbf{C}^{(\ell)} \\ \mathbf{K}^{(\ell)} = \mathbf{W}_{\mathbf{K}}^{(\ell)} \cdot \mathbf{Z}^{(\ell-1)} \\ \mathbf{V}^{(\ell)} = \mathbf{W}_{\mathbf{V}}^{(\ell)} \cdot \mathbf{Z}^{(\ell-1)} \end{cases} \end{aligned} \tag{2}$$

This produces N_ℓ updated part features $\mathbf{Z}^{(\ell)}$ for level ℓ.[1] The benefit of this design is two-fold:

(i) the codebooks act as information bottlenecks, encouraging the learned codes to capture recurring structures across shapes.

(ii) part–subpart relationships in the hierarchy are fully learnable and emerge via the attention matrix $\mathbf{A}^{(\ell)}$.

The matrix $\mathbf{A}^{(\ell)}$ defines a soft adjacency between part features of the previous level and the part codes of the current level. Hence, a single part $\mathbf{p}$ from the previous level can be interpreted to be "assigned" as parent to each subpart $\mathbf{s}$ at the current level as:

$$\mathbf{p} = \arg\max \mathbf{A}_{\mathbf{s},p}^{(\ell)}. \tag{3}$$

To make this discrete parent selection differentiable, we use a straight-through estimator. Specifically, we define a pseudo one-hot vector $\tilde{\boldsymbol{p}} \in \{0,1\}^{N_{\ell-1}}$, that behaves like a hard assignment in the forward pass but preserves gradients from the soft attention:

$$\tilde{\boldsymbol{p}} = \mathbf{A}_{\mathbf{s},\cdot} + \not\nabla \left(\mathbb{1}[\mathbf{p}] - \mathbf{A}_{\mathbf{s},\cdot} \right), \tag{4}$$

where $\not\nabla$ is the stop-gradient symbol, and $\mathbb{1}[\mathbf{p}]$ is a one-hot vector active at index $\mathbf{p}$.

3.2. Geometric part parametrization

With the tree hierarchy defined, we now describe how each recovered part is grounded to 3D geometry. As the following applies identically across decoder levels, in what follows we drop the layer superscript. We take inspiration from CvxNet [9] in parameterizing each part as a 3D convex.

Representing parts as convexes. At each level, we augment the decoder with $\mathcal{G}_\phi$, a set of fully connected layers that map each subpart feature $\mathbf{Z}_\mathbf{s}$ to the parameters of a convex:

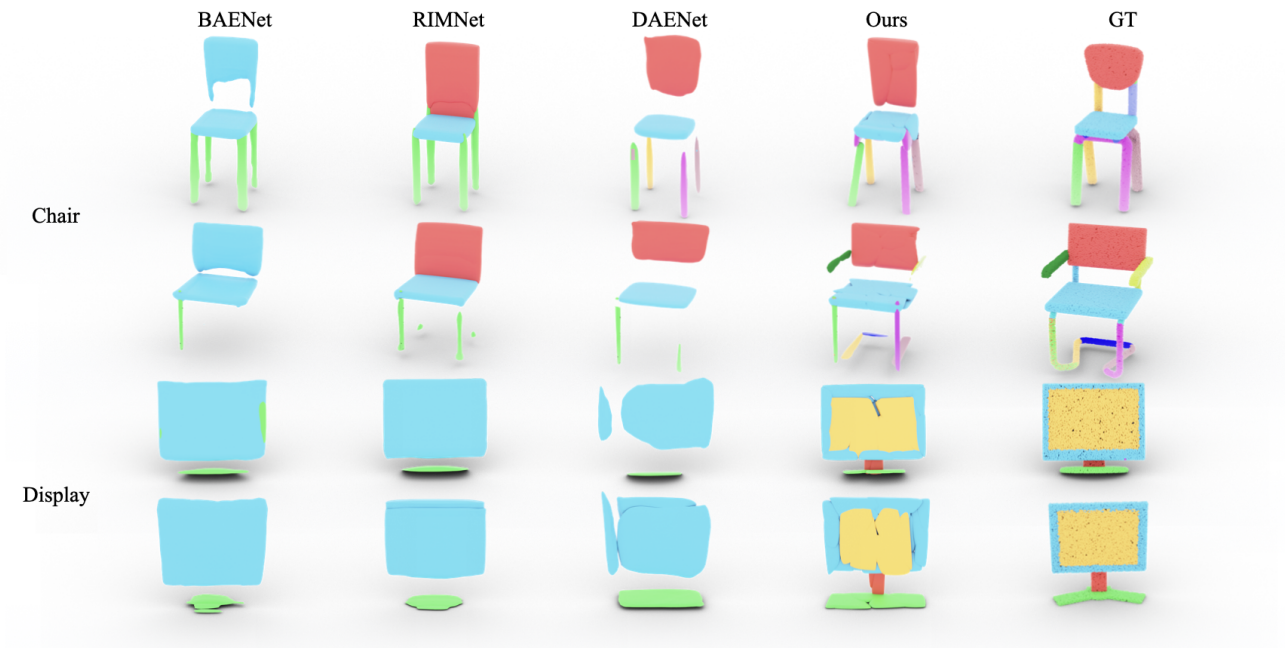
$$\mathcal{C}_\mathbf{s} = \mathcal{G}_\phi(\mathbf{Z}_\mathbf{s}). \tag{5}$$

Each convex $\mathcal{C}_\mathbf{s}$ is defined by H half-spaces, parameterized by plane normals, offsets, and blending weights: $\{\mathbf{n}_\mathbf{s}^h, \mathbf{o}_\mathbf{s}^h, \delta_\mathbf{s}\}_{h=1}^H$. In addition, each convex is assigned a rigid transformation specified by rotation parameters as Euler angles $\mathcal{E}_\mathbf{s}$, translation $\mathbf{t}_\mathbf{s}$, and scale $\mathbf{s}_\mathbf{s}$. The occupancy field for subpart $\mathbf{s}$ is then defined as:

$$\begin{aligned} \tilde{\mathcal{O}}_\mathbf{s}(\mathbf{x}) &= \mathrm{Sigmoid}(-\sigma \Phi_\mathbf{s}(\mathbf{x})), \\ \Phi_\mathbf{s}(\mathbf{x}) &= \log \sum_{h=1}^{H} \exp\left(\mathbf{n}_\mathbf{s}^h \cdot \tilde{x} + \mathbf{o}_\mathbf{s}^h\right), \\ \tilde{x} &= \mathbf{R}(\mathcal{E}_\mathbf{s})^\top \left(\frac{\mathbf{x} - \mathbf{t}_\mathbf{s}}{\mathbf{s}_\mathbf{s}}\right) \end{aligned} \tag{6}$$

where $\tilde{x}$ is the query point transformed into the local coordinate frame of the convex, and σ is a hyperparameter controlling the sharpness of the SDF; We set $\sigma = 75$.

[1] Note that differently from a classical transformer where the number of tokens in the input matches the number of tokens in the output, in our architecture the number of tokens in the output layer matches the cardinality of the codebook within the layer.



Method	Bag	Bed	Bottle	Bowl	Chair	Clock	Washer	Disp.	Door	Earphone	Faucet	Hat	Mean
BAENet [5]	31.7	4.2	32.5	77.1	8.1	16.8	12.8	23.5	27.7	13.8	17.0	39.9	25.42
RIMNet [27]	40.3	9.1	56.2	83.3	18.0	21.2	20.7	37.8	29.5	21.4	25.6	53.9	34.75
DAENet [7]	43.7	12.6	58.1	82.1	31.2	24.2	28.9	41.9	32.7	34.3	36.4	55.4	40.28
Ours	**46.1**	**35.3**	**69.2**	**86.2**	**40.8**	**34.6**	**35.1**	**55.4**	**39.8**	**43.0**	**40.1**	**58.4**	**48.66**

Figure 3. We outperform all baselines in the part segmentation task on ShapeNet, both qualitatively and quantitatively (IoU ↑). Our dynamic tree structure adapts to geometry variations within a category (e.g., chairs), discovering a varying number of parts, while fixed-tree baselines fail to capture such differences.

Containment. Although this formulation associates each subpart in the hierarchy with a geometric primitive, it does not alone enforce spatial consistency with the part–subpart relationships defined by the transformer. To ensure spatial containment of sub-parts in parent parts, we modulate each sub-part's occupancy with that of its parent:

$$\hat{\mathcal{O}}_{\mathbf{s}}(\mathbf{x}) = \hat{\mathcal{O}}_{\mathbf{p}}(\mathbf{x}) \cdot \tilde{\mathcal{O}}_{\mathbf{s}}(\mathbf{x}), \quad \hat{\mathcal{O}}_{\mathbf{p}}(\mathbf{x}) = \sum_{p=0}^{N_{\ell-1}} \tilde{\boldsymbol{p}}_p \cdot \hat{\mathcal{O}}_p(\mathbf{x}) \quad (7)$$

where $\tilde{\boldsymbol{p}}$ is the one-hot vector indicating the parent part for this subpart, defined by (4). This constraint ensures that the subpart has valid nonzero occupancy only when its contained in its parent's spatial support.

3.3. Training objectives

We train our network using a combination of losses: reconstruction of the shape's occupancy field at each level, regularizations on the convex parameters, and structural constraints to maintain a balanced and spatially valid hierarchy:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_1 \mathcal{L}_{\text{contain}} + \lambda_2 \mathcal{L}_{\text{cvxnet}} + \lambda_3 \mathcal{L}_{\text{balance}}. \quad (8)$$

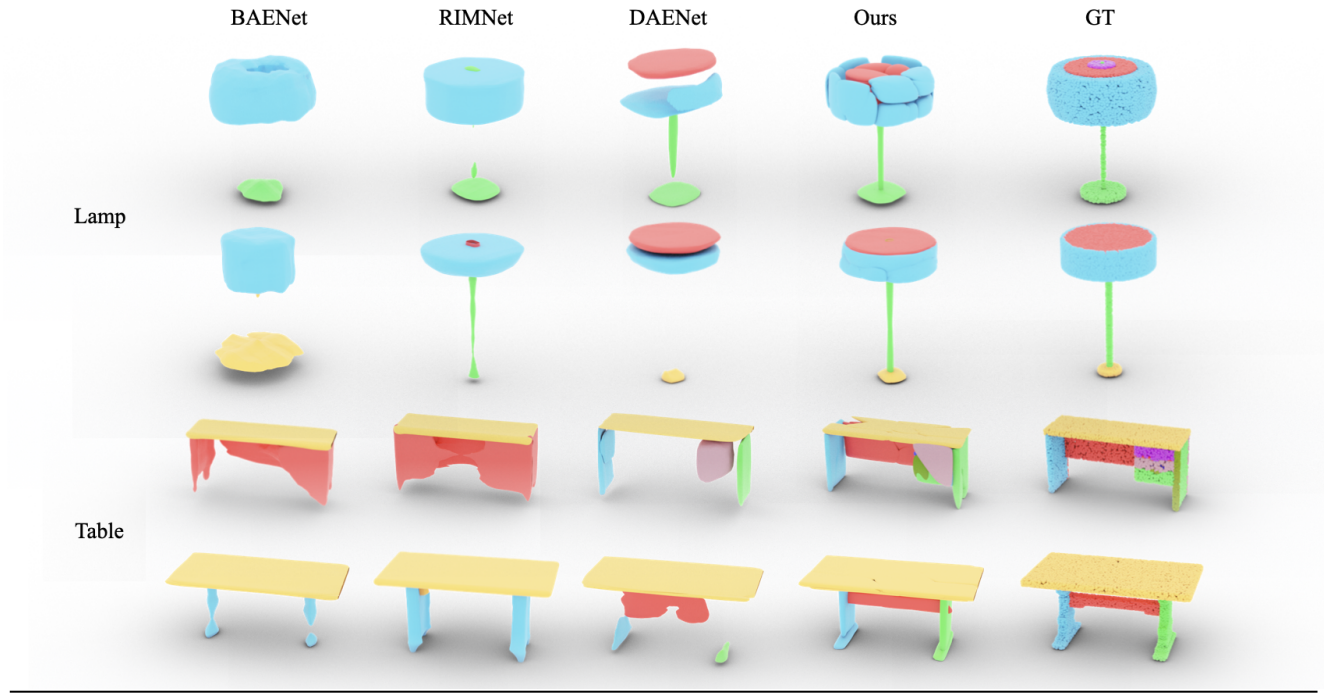
Occupancy reconstruction. The reconstruction loss is applied *per-level*, encouraging the union of part occupancies to best approximate the ground truth occupancy at that level:

$$\mathcal{L}_{\text{recon}} = \sum_{\ell=0}^{L} \left(\mathcal{O}(\mathbf{x}) - \max_p \{ \hat{\mathcal{O}}_p^{(\ell)}(\mathbf{x}) \} \right). \quad (9)$$

While this, together with the containment constraint in (7), encourages sub-parts to lie inside their parent in order to explain the shape occupancy, it does not strictly prevent sub-parts from "bleeding" outside their parent's support. To enforce containment, we define:

$$\mathcal{L}_{\text{contain}} = \sum_{\ell=0}^{L} \sum_{\mathbf{s}=1}^{N_\ell} \left(1 - \hat{\mathcal{O}}_{\mathbf{p}}(\mathbf{x}) \right) \cdot \tilde{\mathcal{O}}_{\mathbf{s}}(\mathbf{x}). \quad (10)$$

Convex regularization. For $\mathcal{L}_{\text{cvxnet}}$, we adopt convex regularizers from CvxNet [9]. Specifically, we use the *decomposition* loss $\mathcal{L}_{\text{decomp}}$ from [9, Eq. 4] to discourage overlapping convexes that redundantly explain the same regions of the shape. We also incorporate their *guidance* loss and a slightly modified locality loss, $\mathcal{L}_{\text{guide}}$ and $\mathcal{L}_{\text{loc}}$, from [9, Eq. 6, 7],



Method	Key	Knife	Lamp	Laptop	Micro	Mug	Fridge	Scissors	Storage	Table	Trash	Vase	Mean
BAENet [5]	0.7	33.9	22.1	16.9	7.6	42.0	10.9	11.7	3.8	9.1	17.6	58.6	19.57
RIMNet [27]	0.8	54.4	35.6	18.6	9.1	42.4	13.3	16.2	11.3	15.9	29.5	68.4	26.29
DAENet [7]	1.0	28.1	38.4	17.7	12.9	56.2	25.7	17.1	21.7	30.1	33.1	65.9	28.98
Ours	**1.7**	**75.8**	**47.1**	**21.8**	**16.5**	**72.4**	**32.2**	**57.9**	**30.0**	**29.9**	**38.5**	**73.3**	**41.43**

Figure 4. Qualitative and quantitative (IoU ↑) results on the ShapeNet dataset show that our method achieves improved part segmentation by accurately reconstructing and consistently recovering recurring parts, whereas baselines often misclassify or miss them entirely.

which discourage the formation of "dead" convexes (i.e., those with near-zero volume and no contribution to shape reconstruction). In particular, we modify $\mathcal{L}_{\text{guide}}$ to make it a *symmetric* (i.e. two-sided) Chamfer distance, additionally minimizing the distance from each query point to its nearest convex center. In combination with our containment constraint (7), the locality and guidance losses prevent sub-parts from collapsing outside their parent. Specifically, under (7), any subpart lying entirely outside its parent will be assigned zero occupancy by construction, and therefore cannot satisfy these two objectives. Thus, these losses help recover such sub-parts by pulling them back into a valid configuration.

Balancing the tree. We encourage more balanced tree structures, where each parent has a non-empty set of children. We do this by minimizing the variance in the number of sub-parts assigned per parent [34]. Letting attention columns represent soft assignments, we write this as:

$$\mathcal{L}_{\text{balance}} = \sum_{\ell=0}^{L} \sum_{p=1}^{N_\ell} \left(\mathbf{s}_p - \frac{1}{N_\ell} \sum_q \mathbf{s}_q \right)^2, \mathbf{s}_p = \sum_{\mathbf{s}=1}^{N_{\ell+1}} \mathbf{A}_{\mathbf{s},p}. \tag{11}$$

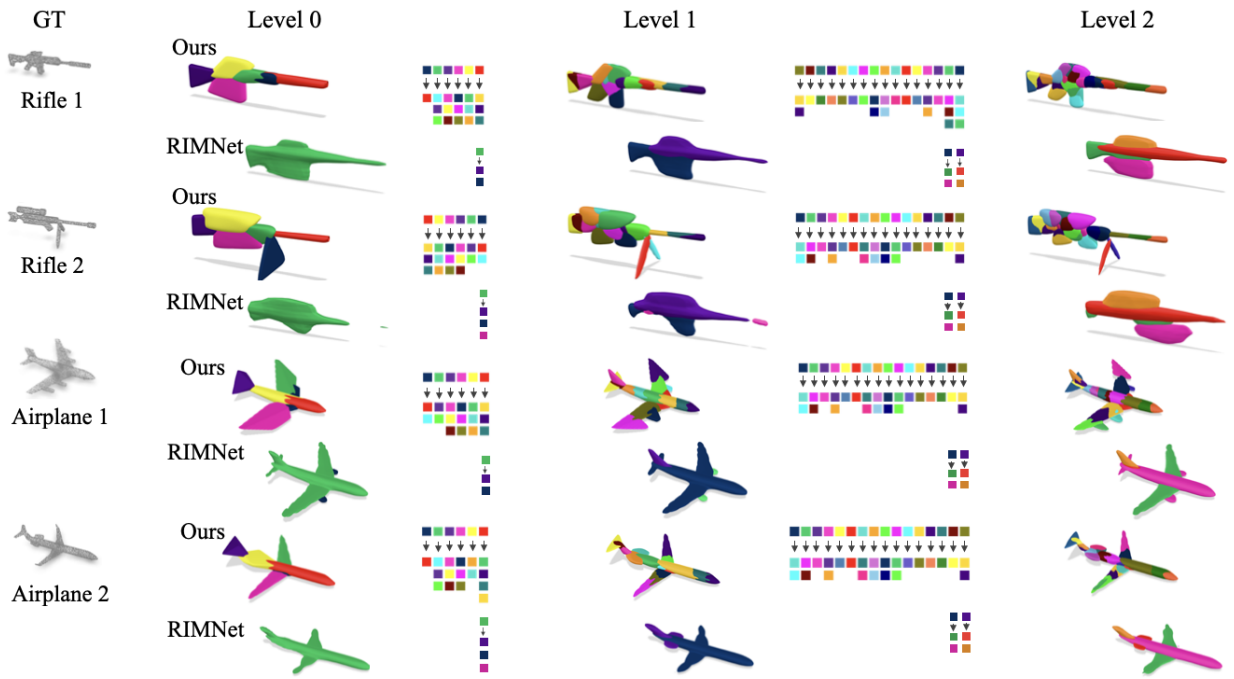
4. Experiments

We validate our method through the task of part segmentation (section 4.1), where we show our method can outperform all previous baselines in recovering consistent parts in all categories. We then show how our hierarchical part reconstruction achieves better reconstruction metrics in finer levels while higher levels remain semantically meaningful level of shape abstraction (section 4.2). Finally, we ablate our design choices and analyze our training dynamics and recovered codebooks (section 4.3).

4.1. Part segmentation – figures 3 and 4

We evaluate our hierarchical transformer on a common part segmentation benchmark and provide comparisons with state-of-the-art single- and multi-level part segmentation methods. We show a significant improvement in segmentation results, while recovering more coherent parts across each class of objects.

Dataset. We train our method on 55 categories from the ShapeNet-v2 dataset, following the train-test split used by Zhang et al. [43]. Our approach supports a unified train-



	CD (↓)				IoU(↑)			
	BAENet	DAENet	RIMNet	Ours	BAENet	DAENet	RIMNet	Ours
Level 0	0.0711	0.0643	0.0661	**0.0630**	0.6167	0.6154	**0.6171**	0.6023
Level 1	-	-	0.0658	**0.0588**	-	-	**0.6192**	0.6178
Level 2	-	-	0.0655	**0.0576**	-	-	0.6181	**0.6242**
Level 3	-	-	0.0663	**0.0576**	-	-	0.6214	**0.6228**

Figure 5. Our hierarchical part segmentation and reconstruction method produces a coherent multi-level shape abstraction: higher levels represent the main structural components, while finer levels capture detailed sub-parts, yielding more accurate reconstructions than prior approaches. The color maps show parent-child relationship between levels.

ing scheme, allowing all categories to be trained jointly. We further analyze this design in section 4.3, where we compare it to category-specific training. For segmentation accuracy evaluation, since the ShapeNet [2] dataset does not provide segmentation labels, we leverage the *fine-grained* part annotations from the PartNet [25] dataset. This differs from previous methods, which evaluate segmentation accuracy on the ShapeNetPart dataset [40], containing only *coarse* segmentation ground-truth labels. The Door, Scissors, and Refrigerator categories appear only in PartNet and not in ShapeNet; hence for these, we use all instances for evaluation only. Each ground-truth point cloud for evaluation in PartNet dataset contains $10,000$ points.

Metrics. We evaluate segmentation performance using average Intersection over Union (IoU) on the segmented point cloud, following prior works [5, 7, 27]. Each ground-truth point in the input point cloud is treated as a query to the predicted convexes at the final level of the hierarchy, and is assigned the label of the convex with the highest occupancy value. Following BSPNet [6], we assign a ground-truth-consistent label to each code in the codebook by identifying the label with the highest number of points falling into the code's corresponding convex. This label-code association is computed once per category on a single instance and then used for all other instances within that category.

Baselines. We compare against BAENet [5], RIMNet [27], and DAENet [7] for part segmentation. BAENet and DAENet perform single-level segmentation, while RIMNet uses a fixed binary hierarchy. For evaluation, we use the predicted parts from the final level of each method. All baselines require voxelized occupancy as input; therefore, we use a CUDA-accelerated voxelizer (binvox) to voxelize the watertight meshes provided in [43]. For fair comparison, we expand the output branching factor of all baselines to 32 to match the number of leaf nodes to our network and train them accordingly. In particular, we train RIMNet for 5 levels to match the total number of leaf nodes with our model.

Analysis. Our method significantly outperforms all baselines in part segmentation, both quantitatively and qualitatively as shown in Fig. 3 and 4. It produces consistent labeling of corresponding parts across instances within a category, whereas baselines often miss parts entirely or misclassify semantically similar regions.

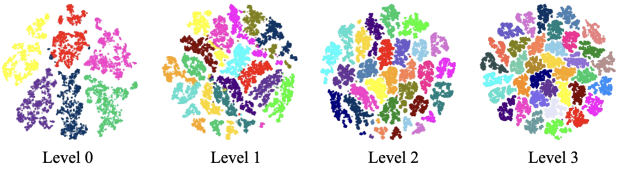
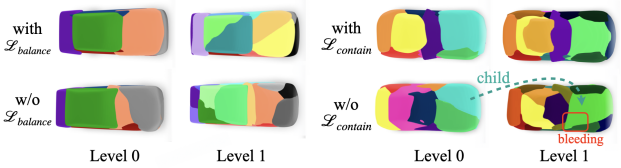


Figure 6. t-SNE visualization of subpart features $\mathbf{Z}^{(\ell)}$ across the ShapeNet test set shows that embeddings for each part (color-coded) form coherent clusters in the embedding space.



Metric \ Loss	All	$-\mathcal{L}_{decomp}$	$-\mathcal{L}_{loc}$	$-\mathcal{L}_{guide}$	$-\mathcal{L}_{contain}$	$-\mathcal{L}_{balance}$
Vol-IoU (↑)	**0.723**	0.669	0.671	0.662	0.719	0.721
CD (↓)	**0.044**	0.063	0.069	0.065	**0.044**	0.045

Figure 7. We show the effect of each of our loss components on part segmentation. (Left) Balance loss helps achieving a more even decomposition. (Right) Addition of containment loss prevents child parts bleeding out of their parent convex.

4.2. Hierarchical reconstruction – figure 5

We demonstrate that our model captures coherent high-level abstractions at early levels and progressively refines fine-grained details at deeper levels. Following the protocol of [43], we train and test on ShapeNet-v2 using the official splits. For evaluation, we compare reconstructed meshes—sampled to 100,000 points each—against ground-truth point clouds, using symmetric Chamfer Distance and voxelized IoU at 128^3 resolution. Our hierarchy employs $[6, 16, 24, 36]$ parts. We compare against BAENet [5], RIMNet [27], and DAENet [7]. While BAENet and DAENet are single-level methods, RIMNet supports hierarchical reconstruction but only via a fixed binary tree, which yields coarse abstractions and loss of detail in later levels (figure 5). By contrast, our model recovers flexible multi-branch hierarchies that balance abstraction and detail across levels.

4.3. Ablation study – table 1 and figures 6 to 8

We ablate our design choices by analyzing the effect of each loss, showing all loss components contribute to the quality of the decomposition. While balance and containment losses have little effect on quantitative metrics, removing them yields imbalanced and non–self-contained hierarchies, as seen in the qualitative results. While our method offers an effective, convenient, and generalized training on all object categories at once, we further analyze how a per-category training similar to previous works [5, 7, 27] affects our results. This shows that a per-category training can result in an improved specialized model for each category, in exchange for more time and compute. We provide a t-SNE analysis of

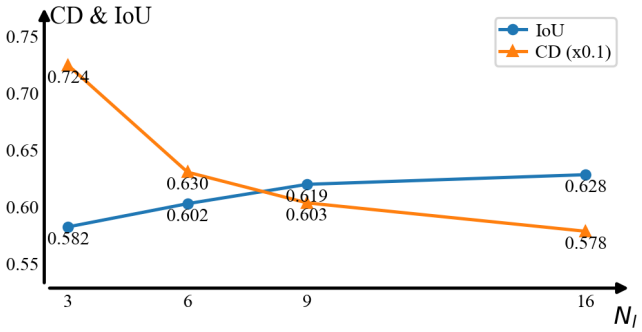


Figure 8. CD and IoU comparison with varying number of parts. As number of parts increases the reconstruction quality improves. N_l is the number of parts at level 0. CD values are scaled by 0.1.

Metric	Method	Categories						Mean
		Bed	Chair	Display	Faucet	Earphone	Lamp	
IoU (↑)	BAENet	5.2	14.9	36.2	16.5	16.7	28.9	19.73
	RIMNet	7.3	19.2	44.8	25.5	16.8	39.7	25.55
	DAENet	14.1	29.9	49.7	37.9	30.4	38.3	33.38
	Ours	**34.6**	**38.8**	**56.2**	**41.3**	**42.7**	**45.3**	**43.15**

Table 1. Our model can also be trained per-category, as opposed to on all shapes. We show results across six ShapeNet categories [2].

the learned part embeddings, demonstrating that embeddings of similarly labeled parts cluster closely together. Finally, in figure 8 we show that reconstruction quality improves as number of parts increases.

5. Conclusion

We introduce HiT, a self-supervised attention-based hierarchical neural field representation that learns general shape abstractions in a coarse-to-fine manner across diverse categories. Our core contribution is a novel cross-attention mechanism which enables dynamic discovery of parent-child relationships across hierarchy levels. To the best of our knowledge, our method is the first to enable general cross-category hierarchical 3D shape abstraction.

However, our current method is constrained by the requirement that the number of parts at each hierarchy level must be fixed. Such a constraint can sometimes lead to unnatural decompositions, particularly at finer levels of the hierarchy, limiting its effectiveness in downstream applications. Recent trends that leverage foundation models as semantic priors offer a promising way to address this limitation, as they enable semantically meaningful grouping of primitives based on high-level prior knowledge. A promising future direction is to make the number of convexes adaptive, for example by selecting them based on sparsity or reconstruction quality, so that the hierarchy remains compact and semantically meaningful.

Acknowledgements. We thank all the anonymous reviewers for their insightful comments. This research was supported by Adobe gift funds.

References

[1] Ahmed Abdelreheem, Ivan Skorokhodov, Maks Ovsjanikov, and Peter Wonka. Satr: Zero-shot semantic segmentation of 3d shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv*, 2015. 7, 8

[3] Siddhartha Chaudhuri, Daniel Ritchie, Jiajun Wu, Kai Xu, and Hao Zhang. Learning generative models of 3d structures. *Computer Graphics Forum (Eurographics STAR)*, 2020. 2

[4] Runnan Chen, Xinge Zhu, Nenglun Chen, Wei Li, Yuexin Ma, Ruigang Yang, and Wenping Wang. Zero-shot point cloud segmentation by transferring geometric primitives. *arXiv*, 2022. 3

[5] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. Bae-net: Branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 2, 5, 6, 7, 8

[6] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 2, 7

[7] Zhiqin Chen, Qimin Chen, Hang Zhou, and Hao Zhang. Dae-net: Deforming auto-encoder for fine-grained shape co-segmentation. In *ACM SIGGRAPH 2024 Conference Papers*, 2024. 2, 5, 6, 7, 8

[8] Dale Decatur, Itai Lang, and Rana Hanocka. 3d highlighter: Localizing regions on 3d shapes via text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[9] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 2, 4, 5

[10] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[11] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 2

[12] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. Pointgmm: A neural gmm network for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[13] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Spaghetti: Editing implicit shapes through part aware generation. *ACM Transactions on Graphics (TOG)*, 2022. 2

[14] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network. *Neural Computation*, 2023. 3

[15] D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 1984. 2

[16] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[17] Juil Koo, Ian Huang, Panos Achlioptas, Leonidas J Guibas, and Minhyuk Sung. Partglot: Learning shape part segmentation from language reference games. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[18] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[19] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 2017. 2

[20] Weiyu Liu, Jiayuan Mao, Joy Hsu, Tucker Hermans, Animesh Garg, and Jiajun Wu. Composable part-based manipulation. In *CoRL 2023*, 2023. 2

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. 3

[22] Björn Michele, Alexandre Boulch, Gilles Puy, Maxime Bucher, and Renaud Marlet. Generative zero-shot learning for semantic segmentation of 3d point clouds. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021. 3

[23] Marvin Minsky. *Society of Mind*. Simon & Schuster, 1986. 2

[24] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 2

[25] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 7

[26] Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models. *arXiv preprint arXiv:2110.13711*, 2021. 3

[27] Chengjie Niu, Manyi Li, Kai Xu, and Hao Zhang. Rim-net: Recursive implicit fields for unsupervised learning of hierarchical shape structures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3, 5, 6, 7, 8

[28] Stephen E. Palmer. Hierarchical structure in perceptual representation. *Cognitive Psychology*. 2

[29] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[30] Despoina Paschalidou, Angelos Katharopoulos, Andreas Geiger, and Sanja Fidler. Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[31] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 2020. 4

[32] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[33] Yaoxian Song, Penglei Sun, Piaopiao Jin, Yi Ren, Yu Zheng, Zhixu Li, Xiaowen Chu, Yue Zhang, Tiefeng Li, and Jason Gu. Learning 6-dof fine-grained grasp detection based on part affordance grounding. *IEEE Transactions on Automation Science and Engineering*, 2025. 2

[34] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. Canonical capsules: Self-supervised capsules in canonical pose. *Advances in Neural information processing systems*, 2021. 6

[35] Shitao Tang, Sicong Tang, Andrea Tagliasacchi, Ping Tan, and Yasutaka Furukawa. Neumap: Neural coordinate mapping by auto-transdecoder for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[36] Konstantinos Tertikas, Despoina Paschalidou, Boxiao Pan, Jeong Joon Park, Mikaela Angelina Uy, Ioannis Emiris, Yannis Avrithis, and Leonidas Guibas. Generating part-aware editable 3d shapes without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[37] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022. 3

[38] Oliver Van Kaick, Kai Xu, Hao Zhang, Yanzhen Wang, Shuyang Sun, Ariel Shamir, and Daniel Cohen-Or. Co-hierarchical analysis of shape structures. *ACM Transactions on Graphics (TOG)*, 2013. 3

[39] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi SM Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. *arXiv preprint arXiv:2111.13260*, 2021. 3

[40] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 2016. 7

[41] Kangxue Yin, Jun Gao, Maria Shugrina, Sameh Khamis, and Sanja Fidler. 3dstylenet: Creating 3d shapes with geometric and texture style variations. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021. 2

[42] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 2

[43] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 2023. 6, 7, 8

[44] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 3