Shape-Based Features Complement CLIP Features and Features Learned from Voxels in 3D Object Classification

Zhi Ji WHITNEY.JI@MAIL.UTORONTO.CA
University of Toronto

Michael Guerzhoy GUERZHOY@CS.TORONTO.EDU
University of Toronto

Abstract

Rezanejad et al. recently showed that symmetry-based contour descriptors improve convolutional neural network (CNN) performance on 2D scene categorization, indicating that complex symmetry-based features cannot necessarily be learned and/or represented with CNNs. In this work, we investigate whether there is evidence for a similar phenomenon in 3D visual data. Using 45,949 object instances from ScanNet spanning 440 classes, we evaluate ten model architectures across fifteen feature sets, including CLIP embeddings, learned features from voxel, and explicitly computed 3D descriptors: geometric statistics and symmetry-based features extracted with SymmetryNet. We observe that explicit geometric and symmetry-based descriptors consistently provide additional predictive information and improve test classification accuracy. We study the possibility of recovering symmetry-based and geometric features from CLIP embeddings, and we show that they are partially recoverable from CLIP features.

Our findings extend Rezanejad et al.'s 2D results to 3D, and further demonstrate that symmetry-based and geometric features provide complementary information beyond foundation model embeddings and raw voxel representations. This provides preliminary evidence that global shape-based features may be useful in open-world 3D scene understanding.

Keywords: 3D features, shape-based features, geometric features, symmetry features, explicit vs implicit representations, 3D object classification

1. Introduction

Geometric descriptors have long played a role in visual recognition, even as deep neural networks have come to dominate the field. Rezanejad et al. [13] showed that 2D symmetry-based contour features derived from medial axis transforms significantly improved CNN scene categorization. Their findings demonstrated that explicit structural descriptors, when combined with learned representations, provide measurable benefit beyond what networks implicitly discover. It remains unclear whether this conclusion extends to 3D, namely whether higher-order 3D shape descriptors provide additional predictive information beyond what modern model architectures already capture.

Recent evidence suggests that explicit computation remains valuable in modern learning. For example, E(3)-equivariant networks achieve orders-of-magnitude sample efficiency in physics-informed modeling [1], underscoring that invariances can dramatically improve training efficiency. Foundation models such as CLIP (Contrastive Language-Image Pre-Training) [12] exhibit strong texture bias [4] and weaknesses in spatial and compositional

reasoning [11; 17]. Vision transformers used in CLIP lack built-in equivariance to rotations and reflections, and positional encodings may further disrupt symmetry. These limitations raise the possibility that explicit geometric and symmetry descriptors could provide additional information that large models do not reliably encode (see Appendix A for extended discussion).

In this work, we extend Rezanejad et al. [13]'s 2D findings into 3D. Using 45,949 ScanNet [2] object instances spanning 440 classes after filtering underrepresented categories, we evaluate explicit geometric statistics and symmetry descriptors from SymmetryNet [16] alongside learned embeddings such as CLIP and voxel-learned features. We construct a systematic grid of 10 architectures and 15 input feature sets combinations (127 conditions total), enabling controlled comparison of explicit versus implicit feature utility.

Our results show that symmetry-based and geometric descriptors consistently provide additional information: classification accuracy improves when these features are included, independent of model family. We also demonstrate that CLIP embeddings contain partially recoverable shape information, but remain less reliable than using explicit descriptors directly. Taken together, these findings extend Rezanejad et al. [13]'s 2D results to 3D, showing that explicit computation of symmetry-based and geometry features can complement foundation model features.

2. Methodology

We explore whether symmetry-based and geometric features improve classification performance, and to what extent those features can be recovered from CLIP embeddings.

First, we conduct a large-scale classification study on ScanNet object instances, systematically evaluating ten model architectures across fifteen feature set combinations, yielding 127 experimental conditions. The results (Figure 1) show how different feature types and model classes contribute to instance-level recognition.

Second, we perform a study where multi-view CLIP embeddings are used to predict explicit symmetry-based and geometric descriptors. This setup explores whether such information can be recovered from vision–language embeddings.

2.1. Input Features

We consider four primary sources of input features: (i) **CLIP embeddings** (512D). Extracted from a frozen ViT-B/32 CLIP image encoder applied to 12 rendered views of each ScanNet object instance. Multi-view embeddings are cached to ensure consistency across experiments. (ii) **Geometric descriptors** (13D). Hand-crafted shape descriptors capturing bounding box aspect ratios, surface-to-volume ratios, and PCA eigenvalue statistics, designed to encode scale- and orientation-independent shape structure. (iii) **SymmetryNet descriptors** (86D). Learned feature embeddings extracted from a pretrained SymmetryNet encoder. These are learned descriptors that are trained to capture reflectional and rotational symmetries of 3D shapes, providing a symmetry-based representation beyond conventional geometry or semantics. We use them as frozen features, without fine-tuning. The SymmetryNet descriptors are computed using the object mesh. (iv) **Voxel occupancy grids.** Binary volumetric representations (R^3) of each instance mesh, processed either directly through a 3D CNN or indirectly as precomputed voxel embeddings (256D).

These inputs are evaluated individually and in concatenated forms, yielding 15 feature sets combinations. Full details of the feature computation are provided in Appendix B, with a summary in Table 1.

2.2. Model Architectures

To probe the interaction between feature type and model class, we instantiate ten model architectures representing four design families (Table 2 in Appendix C): (i) **Linear Baseline:** A single linear projection (CLIPLinear) providing a control for raw feature separability. (ii) **Transformers:** (a) CLIPTransformer, a lightweight 2-layer encoder with learnable [CLS] pooling; (b) FT-Transformer, a feature-token transformer adapted for tabular embeddings. (iii) **Multi-Layer Perceptrons:** Depth-varying MLPs (MLP1–MLP5) with hidden widths $512 - 768$, ReLU activations, and dropout. (iv) **Specialized Models:** (a) MultiModal, a 3-layer fusion MLP for concatenated inputs; (b) VoxCNN, a 3D CNN for raw voxel grids. Since voxel grids are inherently volumetric data, they are only evaluated with VoxCNN, while all other models are designed for tabular or embedding features rather than raw 3D volumes. See Appendix D for training details.

2.3. Complementary Experiments: CLIP $\rightarrow$ Feature Prediction

In addition to classification, we study whether CLIP embeddings encode sufficient geometric and symmetry information to recover explicit descriptors. For each ScanNet object, we render $V = 12$ views and extract frozen CLIP embeddings (512D each), aggregated into a multi-view token sequence. A ViT-style encoder with a learnable [CLS] token produces a global representation, which is mapped via an MLP head to predict either: (i) Symmetry-based descriptors (86D), or (ii) Geometric descriptors (12D). Targets are standardized on the training split, and optimization uses a cosine-augmented mean squared error.

3. Experiments and Results

3.1. Instance-Level Classification Results

Figure 1 presents the test classification accuracy across all model architectures and input feature sets on $45,949$ filtered ScanNet instances (See Appendix E for details). Several clear trends emerge: (i) **Explicit features improve accuracy.** Incorporating geometric and/or symmetry-based descriptors consistently improves classification over CLIP/voxel alone. (ii) **Concatenated features dominate.** The strongest results are obtained when CLIP embeddings are combined with voxel/voxel-derived, geometric and/or symmetry-based descriptors. (iii) **Architectural sensitivity is modest.** While deeper MLPs (MLP2–MLP5) slightly outperform shallower ones, the overall variance across architecture families is smaller than the variance across input feature sets.

These findings demonstrate that explicit geometric and symmetry-based features provide robust additional information for object classification, complementing CLIP embeddings and voxel information.

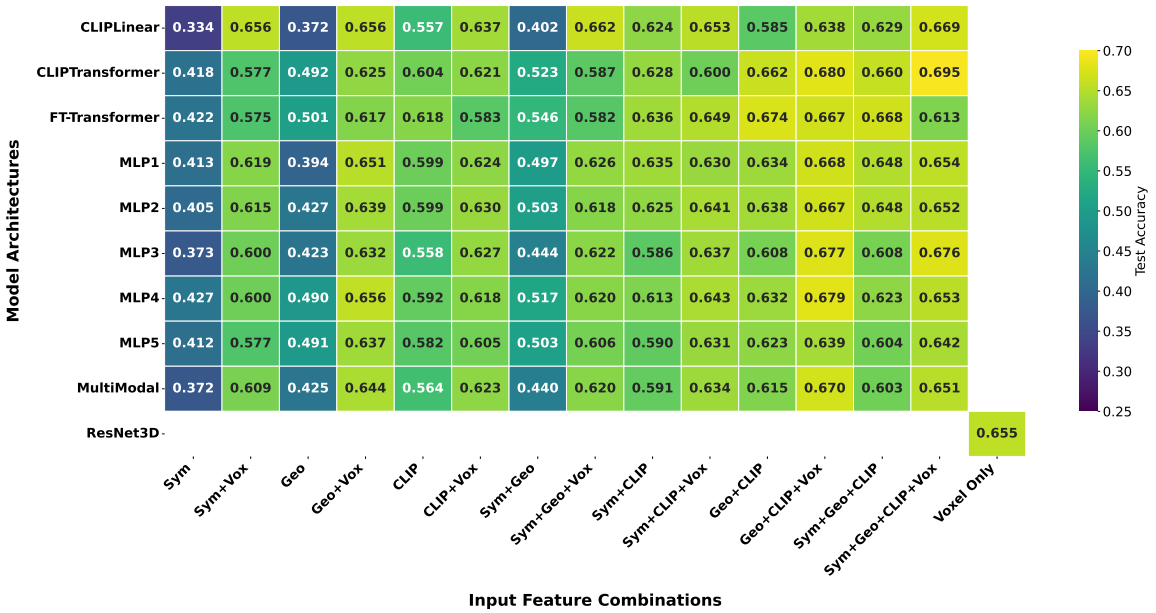

Figure 1: Instance-level classification accuracy on ScanNet across model architectures (rows) and input features (columns).

3.2. CLIP → Shape-Based Features

To probe whether CLIP embeddings implicitly contain shape-based information, we trained a compact ViT-style 3D Transformer to predict symmetry-based and geometric features.

Cosine similarity is ~ 0.75 when predicting geometric features and ~ 0.68 when predicting symmetry-based features, with MSE around 0.4. These results indicate that geometric and symmetry-based descriptors are partially recoverable from CLIP representations.

4. Conclusion

We studied whether higher-order 3D shape descriptors add predictive value beyond what modern model architectures implicitly capture. Across 45,949 ScanNet instances (440 classes), systematic experiments over 10 architectures and 15 input feature sets showed that incorporating geometric or symmetry-based features consistently improves classification performance compared to CLIP embeddings/voxels alone. We show that CLIP embeddings only partially encode geometric and symmetry-based features. These results extend Rezanejad et al. [13]'s 2D findings to 3D, demonstrating that explicit symmetry-based and geometric features complement learned features with additional predictive information. Symmetry-based and geometric features may not be easily learnable or may not be mechanistically representable with our architectures.

Our work provides preliminary evidence for exploring shape-based features in 3D scene understanding. In particular, we provide evidence that as is the case in 2D [13], shape-based features can be useful in 3D as well. Note that the evidence we provide is based on precomputed object meshes. However, shape-based features can be computed for the entire scene as in Rezanejad et al. [13].

References

[1] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1), 2022. doi: 10.1038/s41467-022-29939-5. URL http://dx.doi.org/10.1038/s41467-022-29939-5.

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[3] Peijian Ding, Davit Soselia, Thomas Armstrong, Jiahao Su, and Furong Huang. Reviving shift equivariance in vision transformers, 2023. URL https://arxiv.org/abs/2306.07470.

[4] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness, 2022. URL https://arxiv.org/abs/1811.12231.

[5] Minghao Guo, Bohan Wang, and Wojciech Matusik. Medial skeletal diagram: A generalized medial axis approach for compact 3d shape representation. *ACM Trans. Graph.*, 43(6), November 2024. ISSN 0730-0301. doi: 10.1145/3687964. URL https://doi.org/10.1145/3687964.

[6] Nikolai Kalischek, Rodrigo Caye Daudt, Torben Peters, Reinhard Furrer, Jan D. Wegner, and Konrad Schindler. Biasbed - rigorous texture bias evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22221–22230, June 2023.

[7] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955. doi: 10.1002/nav.3800020109.

[8] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis.*, 2020.

[9] Katelyn Morrison, Benjamin Gilby, Colton Lipchak, Adam Mattioli, and Adriana Kovashka. Exploring corruption robustness: Inductive biases in vision transformers and mlp-mixers, 2021. URL https://arxiv.org/abs/2106.13122.

[10] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=o2mbl-Hmfgd.

[11] Maitreya Patel, Abhiram Kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, and Yezhou Yang. Tripletclip: Improving compositional reasoning of clip via synthetic vision-language negatives, 2024. URL https://arxiv.org/abs/2411.02545.

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[13] Morteza Rezanejad, John Wilder, Dirk B. Walther, Allan D. Jepson, Sven Dickinson, and Kaleem Siddiqi. Shape-based measures improve scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2041–2053, 2024. doi: 10.1109/TPAMI.2023.3333352.

[14] Renan A. Rojas–Gomez, Teck-Yian Lim, Minh N. Do, and Raymond A. Yeh. Making vision transformers truly shift-equivariant. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5568–5577, 2024. doi: 10.1109/CVPR52733.2024.00532.

[15] David W. Romero and Jean-Baptiste Cordonnier. Group equivariant stand-alone self-attention for vision, 2021. URL https://arxiv.org/abs/2010.00977.

[16] Yifei Shi, Junwen Huang, Hongjia Zhang, Xin Xu, Szymon Rusinkiewicz, and Kai Xu. Symmetrynet: Learning to predict reflectional and rotational symmetries of 3d shapes from single-view rgb-d images, 2020. URL https://arxiv.org/abs/2008.00485.

[17] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, 2022.

[18] Duomin Wang, Yu Deng, Zixin Yin, Heung-Yeung Shum, and Baoyuan Wang. Progressive disentangled representation learning for fine-grained controllable talking head synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17979–17989, June 2023.

[19] Pengyuan Wang, Takuya Ikeda, Robert Lee, and Koichi Nishiwaki. Gs-pose: Category-level object pose estimation via geometric and semantic correspondence, 2023. URL https://arxiv.org/abs/2311.13777.

[20] Zehan Wang, Sashuai Zhou, Shaoxuan He, Haifeng Huang, Lihe Yang, Ziang Zhang, Xize Cheng, Shengpeng Ji, Tao Jin, Hengshuang Zhao, et al. Spatialclip: Learning 3d-aware image representations from spatially discriminative language. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29656–29666, 2025.

[21] Renjun Xu, Kaifan Yang, Ke Liu, and Fengxiang He. $e(2)$-equivariant vision transformer, 2023. URL https://arxiv.org/abs/2306.06722.

[22] Zhu Yu, Runmin Zhang, Jiacheng Ying, Junchen Yu, Xiaohai Hu, Lun Luo, Si-Yuan Cao, and Hui-Liang Shen. Context and geometry aware voxel transformer for semantic scene completion, 2024. URL https://arxiv.org/abs/2405.13675.

[23] Yihan Zeng, Chenhan Jiang, Jiageng Mao, Jianhua Han, Chaoqiang Ye, Qingqiu Huang, Dit-Yan Yeung, Zhen Yang, Xiaodan Liang, and Hang Xu. Clip^2: Contrastive language-image-point pretraining from real-world point cloud data, 2023. URL https://arxiv.org/abs/2303.12417.

Appendix A. Background and Related Work

A.1. Explicit descriptors and equivariance.

Classical shape analysis shows that explicit structural cues can add predictive value: symmetry-based contour descriptors improve CNN scene categorization from contours in 2D [13]. In parallel, E(3)-equivariant networks achieve large gains in data efficiency by hard-wiring geometric symmetries into the architecture [1]. Together, these lines of work suggest that explicitly representing geometry—either as features or as constraints—can complement end-to-end learning [18].

A.2. From 2D contours to 3D structure.

Transitioning from 2D outlines to 3D shapes introduces pose, occlusion, and volumetric effects that challenge appearance-only pipelines. Explicit 3D symmetry-based descriptors (e.g., SymmetryNet for reflectional symmetries [16]) and compact geometric statistics (e.g., bounding-box ratios, surface/volume surrogates, PCA eigen-structure) provide pose-robust summaries of shape organization. These descriptors are complementary to voxelized occupancy or learned embeddings: the former capture stable global regularities (axes, planes, repetitions), while the latter excel at semantics but may conflate geometry with texture [5; 8; 19; 22].

A.3. Limits of foundation models for geometric reasoning.

CLIP [12] delivers strong transferable semantics, yet vision models trained on natural images exhibit pronounced texture bias relative to human shape bias [4]. CLIP-style ViT towers also struggle on composition/spatial tests such as Winoground [17], and targeted objectives (e.g., TripletCLIP) improve but do not eliminate these gaps [11]. Moreover, standard ViTs lack built-in rotation/reflection equivariance and positional encodings can disrupt symmetry, motivating the use of explicit geometric signals alongside learned features [3; 6; 9; 10; 14; 15; 20; 21; 23].

A.4. Gap and our focus.

While there is evidence that explicit geometry helps in 2D [13] and that equivariant designs boost data efficiency [1], there has not been a systematic 3D study that tests *across architectures and input sets* whether higher-order geometric and symmetry features provide *additional* predictive information beyond modern learned embeddings. Our work fills this

gap via an apples-to-apples evaluation on ScanNet that pairs explicit 3D descriptors (geometry and symmetry) with CLIP embeddings and voxel-learned features, quantifying their complementarity in 3D recognition.

Appendix B. Inputs

We evaluate 15 input types: 1 voxel-only representation, 7 purely tabular variants based on symmetry, geometry, and CLIP features, and 7 fusion variants combining voxel embeddings with tabular features, as shown in the Table 1.

B.1. Symmetry Features (86D)

SymmetryNet is a deep neural network that predicts both reflectional and rotational symmetries of 3D objects from single-view RGB-D images, addressing the fundamental challenge where severely incomplete data renders traditional geometric approaches infeasible. The architecture employs a point-based processing pipeline that extracts appearance features via CNN and geometric features via PointNet, then fuses these through a multi-task learning framework that jointly predicts symmetry parameters (plane normal and point for reflection, axis direction and point for rotation) alongside dense symmetric correspondences for each 3D point. This multi-task design prevents overfitting by forcing the network to understand local symmetry correspondences rather than memorizing global shape patterns, while an optimal assignment mechanism using the Hungarian algorithm [7] enables detection of multiple symmetries without requiring predefined ordering. The system incorporates visibility-based verification that validates predictions against 3D geometry while accounting for occlusion patterns inherent in single-view observations, and demonstrates robust generalization across unseen object instances, novel categories, and real-world RGB-D data from ShapeNet, YCB, and ScanNet datasets, significantly outperforming geometric fitting baselines that fail entirely on incomplete data.

We are using the 86-dimensional from the pretrained SymmetryNet model [16], which is designed to detect and encode global symmetry patterns in 3D objects. SymmetryNet identifies reflective planes, rotational axes, and translational repetition groups, and encodes them into a fixed-length representation. The resulting feature vector captures fine-grained structural regularities such as bilateral symmetry in furniture, radial symmetry in round objects, or translational patterns in repetitive structures. These symmetries are often difficult to recover implicitly from voxel or image embeddings.

B.2. CLIP Embeddings (512D)

CLIP (Contrastive Language Image Pre Training) is a neural network trained on large collections of image text pairs to learn a shared embedding space. Its image encoder produces vectors that capture high level semantic content, making it a strong general purpose visual feature extractor. Semantic features are provided by frozen CLIP embeddings extracted with the ViT-B/32 image encoder. Each ScanNet instance is rendered from 12 viewpoints, and the resulting image embeddings are cached and standardized to ensure reproducibility. A 512-dimensional vector is used for each object instance.

B.3. Geometric Descriptors (13D)

We compute a set of 13 handcrafted geometric statistics from the 3D mesh of each ScanNet object. These descriptors summarize coarse but informative aspects of object shape and are designed to be invariant to global scale and orientation. Specifically, the 13D vector includes:

- **Bounding box ratios (3D).** Ratios of side lengths (longest/shortest, longest/middle, middle/shortest), capturing elongation and aspect.
- **Volume and surface statistics (2D).** Surface area to volume ratio, and normalized volume relative to the bounding box.
- **Principal component eigenvalues (3D).** Ratios of eigenvalues from PCA on the vertex coordinates, encoding overall spread along principal axes.
- **Compactness and sphericity (2D).** Compactness ($V^{2/3}/A$) and sphericity ($\pi^{1/3}(6V)^{2/3}/A$), measuring deviation from spherical form.
- **Other normalized ratios (3D).** Width/height, depth/height, and depth/width, capturing additional aspect relationships.

This compact set of descriptors provides a stable, interpretable summary of coarse object shape.

B.4. Concatenated Tabular Features

In addition to individual modalities, we construct concatenated tabular inputs to test complementarity:

- symmetry (86D) + geometry (13D) → **99D**,
- geometry (13D) + CLIP (512D) → **525D**,
- symmetry (86D) + CLIP (512D) → **598D**,
- symmetry (86D) + geometry (13D) + CLIP (512D) → **611D**.

Together with the individual sources, these yield seven tabular feature sets in total.

B.5. Voxel Occupancy Grids

Each object mesh is voxelized into a binary occupancy grid of resolution 32^3. This volumetric representation is processed by a 3D ResNet backbone (`r3d_18` by default, pretrained on Kinetics-400). The single-channel grid is repeated across three channels, with the depth dimension treated as time. The backbone applies 3D convolutions followed by global average pooling and a linear head.

Table 1: Summary of input feature types used in our ScanNet experiments. Non-voxel features are standardized to zero mean and unit variance. A pre-trained 3D ResNet backbone is used for voxel-related inputs.

Input	Dim.	Constituents	Source / Description
clip	512	CLIP embeddings	Frozen CLIP ViT-B/32 on multi-view renders.
geometric	13	geometry descriptors	Bounding-box ratios, surface/volume stats, PCA eigenvalue ratios, etc.
symmetrynet	86	SymmetryNet features	Symmetry feature vector from SymmetryNet.
geo_clip_concat	$13 + 512 = 525$	geometric + CLIP	Concatenation of geometric descriptors with CLIP embeddings.
sym_clip_concat	$86 + 512 = 598$	symmetry + CLIP	Concatenation of symmetrynet and CLIP embeddings.
sym_geo_concat	$86 + 13 = 99$	symmetry + geometric	Concatenation of symmetrynet and geometric features.
sym_geo_clip_concat	$86 + 13 + 512 = 611$	symmetry + geometric + CLIP	Concatenation of symmetrynet, geometric descriptors, and CLIP.
voxel	32^3 grid	raw voxel grid	End-to-end 3D ResNet on raw occupancy volumes.
geometric_vox_direct_concat	$13 + 512 = 525$	geometric + voxel emb	Fusion: geometric + ResNet3D backbone embedding.
symmetrynet_vox_direct_concat	$86 + 512 = 598$	symmetry + voxel emb	Fusion: symmetry + ResNet3D embedding.
clip_vox_direct_concat	$512 + 512 = 1024$	CLIP + voxel emb	Fusion: clip + ResNet3D embedding.
sym_geo_vox_direct_concat	$99 + 512 = 611$	(sym+geo) + voxel emb	Fusion: `sym_geo_concat` + ResNet3D embedding.
sym_clip_vox_direct_concat	$598 + 512 = 1110$	(sym+CLIP) + voxel emb	Fusion: `sym_clip_concat` + ResNet3D embedding.
geo_clip_vox_direct_concat	$525 + 512 = 1037$	(geo+CLIP) + voxel emb	Fusion: `geo_clip_concat` + ResNet3D embedding.
sym_geo_clip_vox_direct_concat	$611 + 512 = 1123$	(sym+geo+CLIP) + voxel emb	Fusion: `sym_geo_clip_concat` + ResNet3D embedding.

B.6. Voxel-Tabular Fusion Features

To integrate volumetric and tabular information, a voxel embedding of dimension **512** (the default `vox_emb_dim`) is obtained from the 3D ResNet backbone and concatenated with tabular inputs. The resulting feature sets are:

- geometry (13D) + voxel (512D) → **525D**,
- symmetry (86D) + voxel (512D) → **598D**,
- CLIP (512D) + voxel (512D) → **1024D**,
- (sym+geo) 99D + voxel (512D) → **611D**,
- (sym+CLIP) 598D + voxel (512D) → **1110D**,
- (geo+CLIP) 525D + voxel (512D) → **1037D**,
- (sym+geo+CLIP) 611D + voxel (512D) → **1123D**.

Summary. Across all settings, we study **15** input types: one voxel-only, seven tabular variants, and seven voxel–tabular fusion variants. This design enables systematic evaluation of semantic, geometric, and symmetry-based features in isolation and in combination with volumetric representations.

Appendix C. Models

Model architectures are shown in Table 2.

"Tabular" inputs include `clip`, `geometric`, `symmetrynet`, and all of their concatenations, as well as precomputed voxel CNN embeddings (`voxcnn_emb.npy`). "Voxel grid (raw)" refers to occupancy volumes (`voxel.npy`/`vox.npy`).

Table 2: Model architectures used in our 3D experiments.

Model	**Architecture type**	**Core design / depth**	**Inputs**
CLIPLinear	Linear classifier	Single fully connected layer (`LinearHead`) to logits; dropout 0.10	Tabular
CLIP Transformer	TinyTransformer	Project to $d = 192$; prepend learnable [CLS]; 2 encoder layers ($n_{\text{head}} = 6$, FF=384); dropout 0.10	Tabular
FT-Transformer	Feature-token Transformer	Tokenize to $d = 256$; [CLS] pooling; 2 encoder layers ($n_{\text{head}} = 8$, FF=512); dropout 0.10	Tabular
MLP1	Fully connected (ReLU, Dropout)	Depth = 1; hidden = 512; dropout 0.10	Tabular[1]
MLP2	Fully connected (ReLU, Dropout)	Depth = 2; hidden = 640; dropout 0.10	Tabular[1]
MLP3	Fully connected (ReLU, Dropout)	Depth = 3; hidden = 768; dropout 0.10	Tabular[1]
MLP4	Fully connected (ReLU, Dropout)	Depth = 4; hidden = 768; dropout 0.10	Tabular[1]
MLP5	Fully connected (ReLU, Dropout)	Depth = 5; hidden = 768; dropout 0.10	Tabular[1]
MultiModal	MLP for tabular concatenations	Depth = 3; hidden = 768; dropout 0.10	Tabular[1]
ResNet3D	3D ResNet backbone	Pretrained `r3d_18` (default; or `mc3_18`); input voxels 32^3 with depth as time; 1→3 channel repeat; global avg pool → linear head	Voxel (raw)

[1] For any `*_vox_direct_concat` column, the same heads (Linear/Transformer/MLP) are used but preceded by a ResNet3D backbone. The voxel grid is encoded to a **512**-D embedding (`vox_emb_dim`=512), concatenated with tabular features, and trained end-to-end.

Appendix D. Training Protocol

All classification experiments are conducted on 45,949 ScanNet object instances spanning 440 classes, using a train:validation:test split with a ratio of 8:1:1, after filtering out under-represented classes (classes with very few instances). Optimization settings are fixed across architectures to ensure comparability: AdamW optimizer with learning rate 10^{-5}, weight decay 10^{-4}, and gradient clipping at 1.0. We employ mixed-precision training (AMP) and early stopping based on validation accuracy (patience = 12 epochs, $\Delta_{\min} = 10^{-4}$). For models using CLIP features, the CLIP encoder is frozen; only the downstream classifier is trained.

Appendix E. Dataset Summary

This appendix documents the ScanNet object instance data used in our experiments. The raw cache contains 45,949 instances spanning 551 raw class IDs (IDs 0–550). After filtering out classes with fewer than two instances, 440 valid classes remain. The distribution is highly imbalanced: a small number of categories dominate, while the majority occur rarely.

Raw Distribution. Figure 2 shows the histogram of all 551 classes (log scale). A long-tailed distribution is evident, with many classes occurring fewer than 10 times. See `class_distribution_raw.csv` (available at https://anonymous.4open.science/r/NeurReps_supplementary) for the distribution of all classes.

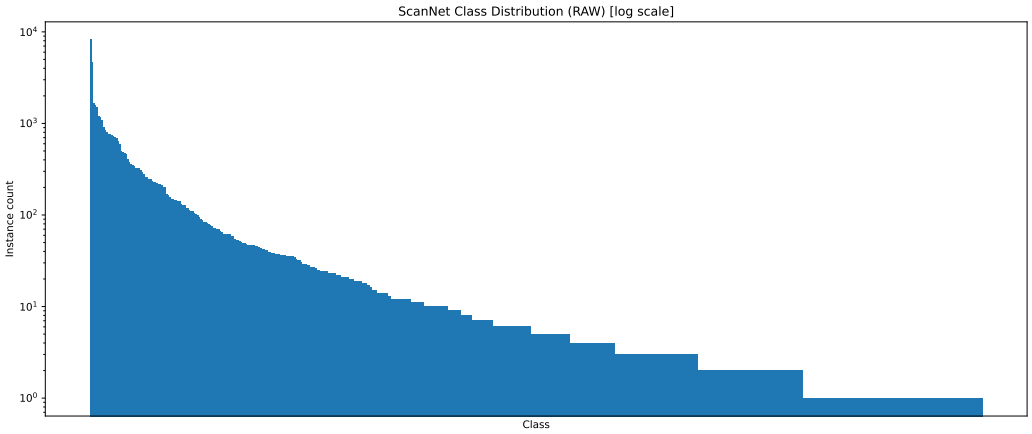

Figure 2: Original class distribution of all 551 classes (log scale).

Filtered Distribution. A stricter filtering threshold (e.g., removing under-represented classes below τ instances) was used. After removing singleton classes, 440 valid categories left. This step removes rare outliers while preserving the bulk of the dataset. See `kept_classes_ge2.csv` (available at https://anonymous.4open.science/r/NeurReps_supplementary) for the distribution of classes after filtered.

Figure 3 presents the most frequent 30 categories after filtering. The largest categories include *wall* (8199 instances), *chair* (4618), *books* (1645), *floor* (1551), and *door* (1475). These dominate the dataset distribution and provide the strongest training signal.

Summary. In total, our experiments use 45,949 instances across 440 valid categories. The dataset's strong imbalance motivates systematic evaluation across both dominant and rare classes.

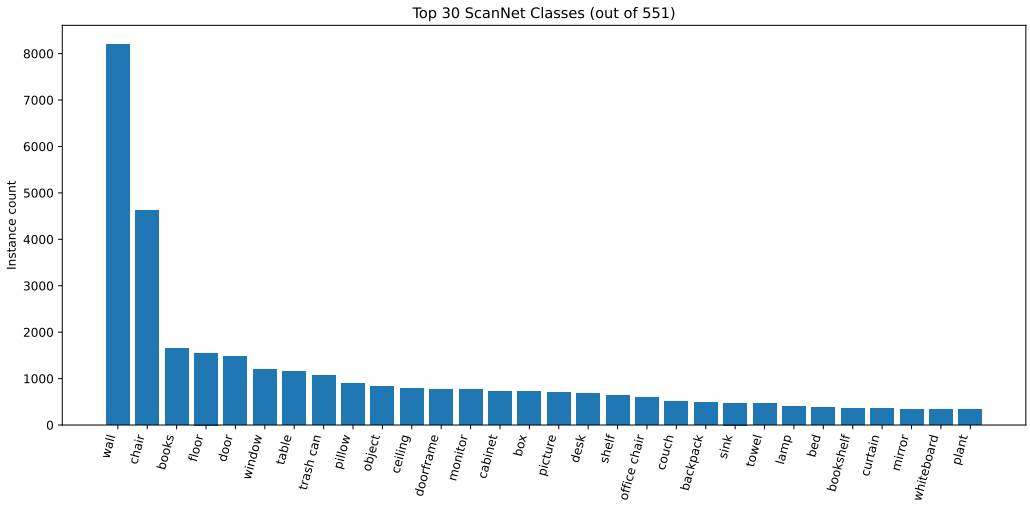

Figure 3: Top 30 most frequent classes after filtering.