Appendix

1. Implementation Details

1.1. pretrain

During the pre-training phase, the model architecture employs a specially designed Hierarchical HeteroGNN with a hidden dimension of 128, comprising 1 hypergraph convolution layer and 2 heterogeneous graph Transformer layers. Each Transformer layer utilizes 4 attention heads and is equipped with 256-dimensional edge feature encoding. The training process adopts the Adam optimizer with a learning rate of 0.0001 and weight decay of 0.00001, processing 8192 graphs per batch on a single GPU for up to 800 training epochs. The random seed is fixed at 42 to ensure complete experimental reproducibility.

For feature encoding, the node encoder processes 8×32×32 geometric grid features through a CNN network and combines them with 17-dimensional semantic attributes to produce 256-dimensional fused features. The edge encoder employs a 1D-CNN to handle 12×32 geometric sampling features along with 12-dimensional semantic attributes, similarly generating 256-dimensional edge representations.

The contrastive learning module sets the temperature parameter to 0.1 and maps 128-dimensional graph-level embeddings to the contrastive space through a two-layer MLP projection head. The key reinforcement learning controller, based on an Actor-Critic architecture and using an independent Adam optimizer (learning rate 0.0001), dynamically generates perturbation strengths ranging from 0.05 to 0.2 according to the current graph embedding state, updating its policy every 5 training batches with reward signals derived from the negative smoothed contrastive loss. The training process incorporates an early stopping mechanism that terminates training prematurely when the smoothed contrastive loss shows no improvement for 20 consecutive epochs.

1.2. Structure-Aware Prompt

In the downstream few-shot classification task, we first load the pre-trained checkpoint and freeze all pre-trained components, including the Hierarchical HeteroGNN backbone (with 128 hidden dimensions and 2 layers of heterogeneous graph Transformers) and all feature encoders. The prompt tuning process employs an alternating optimization strategy: the answering head is optimized with a higher learning rate of 0.01 and weight decay of 2e-6, while the structure-aware prompt module uses a lower learning rate of 1e-6 to maintain stability. The training process is conducted on a single GPU for 400 epochs with a batch size of 16, and the random seed is fixed at 42 to ensure reproducibility.Training employs the cross-entropy loss function and incorporates an early stopping mechanism (patience of 10) to prevent overfitting.

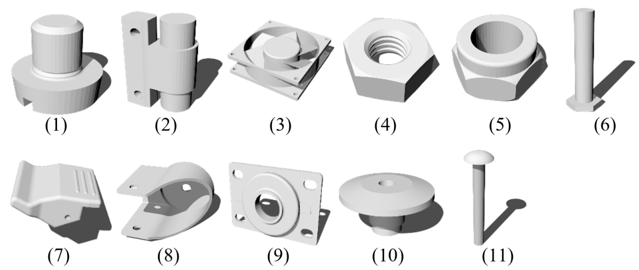

Figure 1. TraceParts-11.

2. Hybrid Graph Representation Detail

2.1. Regional Association Hypergraph Detail

Ball Query. Ball Query serves as the initial hyperedge generation step based on spatial proximity. In the specific implementation, we first construct a KD-tree of all face centroid points to accelerate spatial searches. For each seed point selected through farthest point sampling, we perform a range query centered at that point with an adaptive radius as the search radius. This query returns the indices of all face nodes falling within the spherical region, forming initial "geometric clusters." The adaptive radius calculation comprehensively considers both local point density and global distribution through the formula $r_{adaptive} = (1 - c) \cdot r_{min} + c \cdot r_{max}$, where r_{min} represents the distance from the current seed point to its nearest neighboring seed, r_{max} represents the distance to the farthest seed, and the balance coefficient c is set to 0.1. The raw hyperedges produced by this step are based solely on geometric proximity and may contain topologically disconnected nodes.

Topological Consistency Correction. Topological Consistency Correction is a crucial step to address the issue of geometric proximity coupled with topological isolation. For each initial hyperedge obtained through ball query, we extract the corresponding subgraph from the original adjacency graph and compute the connected components of this subgraph. Only the largest connected component is retained, while the remaining nodes are marked as ”to be relocated.” This operation ensures that all faces within each hyperedge are directly or indirectly connected in terms of topological relations, which aligns with the prior knowledge that functional features (such as holes and protrusions) in B-rep models are typically composed of contiguous faces. The corrected hyperedges possess clear topological significance, laying the foundation for subsequent feature-level representation learning.

Full Adjacency Closure Inference. Full Adjacency Closure Inference serves as the final step for hypergraph refinement, handling boundary cases through an iterative algorithm. The specific procedure is as follows: traverse all nodes not contained by any hyperedge and examine the hyperedge affiliations of their adjacent nodes. If all adjacent nodes of a given node belong to the same hyperedge, that node is incorporated into that hyperedge. This process iterates until no new nodes are added to any hyperedge . This mechanism effectively resolves the ”boundary node assignment” problem, ensuring the integrity of functional units. Particularly for faces located at feature boundaries that are geometrically relatively isolated, it enables correct classification based on their topological context.

2.2. Multi-Semantic Geometry Feature Encoding

Face feature encoding. Face feature encoding integrates 17-dimensional parametric attributes with 8×32×32 non-parametric UV grids to form a comprehensive geometric representation. The parametric features are subdivided into four semantically distinct groups: surface type employs 5-dimensional one-hot encoding to precisely identify fundamental surface types including planes, cylinders, cones, spheres, and tori; area attribute is a 1-dimensional scalar value computed using $BRepGProp_Surface Properties$; structural attributes form a 3-dimensional vector comprising face degree (number of adjacent edges), loop count (sum of outer and inner loops), and adjacent face count; differential-geometric properties provide 7-dimensional curvature features including absolute values of Gaussian curvature, mean curvature, and maximum and minimum principal curvatures. This grouped encoding strategy ensures the preservation of multi-level semantic information ranging from macroscopic types to microscopic geometric characteristics.

Adjacency edge feature encoding. Adjacency edge feature encoding extracts 12-dimensional parametric attributes for topologically connected edges, employing a specialized semantic grouping strategy. The convexity feature uses 3-dimensional one-hot encoding to distinctly differentiate three geometric relationships: convex edges, concave edges, and smooth edges; the length attribute is a 1-dimensional scalar, precisely calculated using $BRepGProp_Linear Properties$ to determine the curve length of the edge; the curve type employs 8-dimensional one-hot encoding to comprehensively cover major curve types including lines, circular arcs, ellipses, hyperbolas, parabolas, Bézier curves, B-spline curves, and offset curves. This fine-grained encoding scheme accurately captures both the geometric properties and functional roles of edges within B-rep models.

Parallel edge encoding. Parallel edge encoding is specifically designed to capture long-range geometric constraints, extracting 8-dimensional semantic relationship features. These features include: the cosine of the angle between normal vectors, reflecting directional similarity between faces; the distance between planes, characterizing spatial positional relationships; the area ratio, describing the size proportion between faces; the centroid distance, providing spatial distribution information; a direction consistency flag, indicating whether normal vectors are aligned in the same or opposite directions; and a 3D offset vector, completely describing the relative spatial position. All extracted features are mapped to a unified 256-dimensional representation space through specialized encoding networks, providing rich geometric and topological prior knowledge for downstream tasks and supporting complex structural understanding and reasoning tasks.

2.3. RL-Enhanced Contrastive Pre-training

During the pre-training process, we employ a two-stage alternating optimization strategy. The first stage focuses on contrastive learning optimization of the GNN encoder: initially, an adaptive perturbation strength is generated by the frozen RL controller, followed by the use of in-place perturbation techniques to create augmented views. The InfoNCE loss between the anchor view and augmented views is then computed, and gradients are backpropagated solely through the anchor view to update GNN parameters, ensuring training stability.

The second stage optimizes the RL controller at a lower frequency: using the exponentially moving averaged contrastive loss as the reward signal and adopting the Advantage Actor-Critic algorithm to simultaneously optimize both the policy network and value network. The controller learns to generate appropriate perturbation strengths that maintain the contrastive learning task at a moderate difficulty level, thereby enhancing the quality of pre-trained representations. The gradient flows of the two stages remain completely separated to prevent mutual interference, ultimately achieving synergistic optimization between GNN representation learning and enhancement strategies.

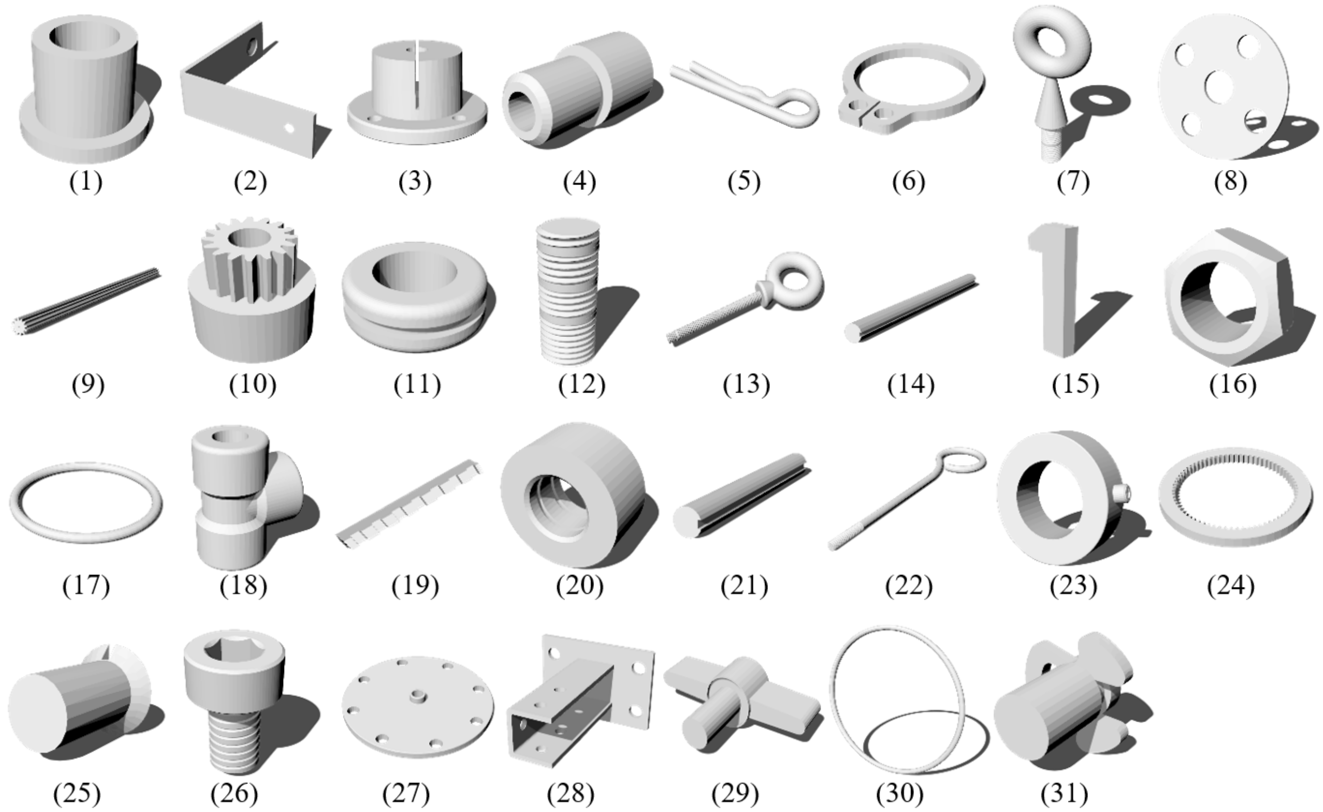

Figure 2. Fabwave-31.

3. Dataset Detail

3.1. Tracepart-11 Dataset

The TraceParts-11 dataset is constructed based on the TraceParts standard parts library, forming 11 more discriminative part categories by performing a fine-grained partitioning of the original 6 ”assembly categories,” totaling 943 models. This fine-grained partitioning strategy enhances the challenge of the classification task, requiring the model to capture more subtle geometric feature differences. Visualizations of representative models from the dataset are provided in Figure 1.

3.2. Fabwave-31 Dataset

The FabWave-31 dataset is derived from the FabWave benchmark library, from which we selected 31 representative mechanical part categories, totaling 2,775 B-rep models. This dataset covers a wide range of industrial manufacturing part types, providing a diverse test benchmark for few-shot classification tasks. Each category in the dataset contains a sufficient number of samples, ensuring the reliability of training and evaluation. Visualizations of representative models from the dataset are provided in Figure 2.