BYOP PROPOSAL

Project Domain: Computer Vision, Generative AI, Geometry Processing

DIVYANSH GARG (24115060) Electrical Engineering

9582025287 divyansh_g@ee.iitr.ac.in

Project Title:

"Learning Editable 3D Neural Shape Priors for Semantic and Constrained Shape Editing"

1. Project Introduction:

3D generative models have made significant progress in recent years, especially with implicit neural representations such as **DeepSDF**, **Occupancy Networks**, and emerging **3D diffusion models**. These models can accurately reconstruct and generate complex shapes.
However, almost **none** of them support **semantic, parametric editing—**a feature essential in CAD modelling, robotics, and industrial design.

In traditional CAD environments, designers can modify specific dimensions (like *handle width*, *seat height*, or *curvature*). In contrast, latent vectors learned by neural networks are typically **entangled**, meaning small modifications cause global distortions.
This project attempts to solve a clear research problem:

"How can we learn editable neural shape priors that support clean, localized, and semantically meaningful modifications?"

This is a technically challenging, underexplored area combining **geometry processing**, **implicit representations**, **latent space learning**, and **differentiable constraints**.

This is technically challenging because it requires combining:

- **Implicit neural representations (SDF/Occupancy models)**
- **Geometry processing and mesh reasoning**
- **Latent space disentanglement**
- **Differentiable constraints (self-intersection, smoothness)**
- **Optimization-based latent editing**

The goal is to transform static neural shape models into **editable, controllable, designer-friendly 3D models** which current research rarely addresses.

2. Inspiration:

This project is inspired by the limitations of existing 3D generative research. While papers like **DeepSDF (Park et al., CVPR 2019)** and **Occupancy Networks (Mescheder et al., CVPR 2019)** successfully represent 3D geometry as continuous fields, they treat the entire shape as a single latent code. This means that any change to the code typically affects the entire shape, making local or semantic edits extremely difficult.

Similarly, recent works such as **Neural 3D Morphable Models**, **Latent3D Diffusion**, and **Neural Parts (CVPR 2021)** show promise in generating diverse shapes but still lack **consistent part structure** and **editability**.

Real-world applications CAD modelling, industrial design, household robotics require **parametric shape editing**: the ability to modify geometry predictably. Current models cannot do this.

This project draws inspiration from three research directions:

- **Neural Implicit Fields** – for continuous 3D representation
- **Part-Based Decomposition** – inspired by semantic mesh segmentation and Neural Parts
- **Differentiable Optimization** – inspired by differentiable rendering (IDR, NeuS) and differentiable physics (MeshGraphNets)

The intersection of these fields is still underdeveloped.
This project attempts to unify them in a simple but technically rigorous way.

3. Proposed Solution :

The proposed solution consists of **four interconnected modules**:

A. Neural Shape Prior (Base Implicit Representation)

I will train an implicit SDF network similar to DeepSDF:

- **Input:** latent vector z + 3D coordinates (x, y, z)
- **Output:** signed distance value

The model learns a **continuous representation** of the shape class (e.g., mugs, chairs).
This becomes the foundation for editing.

Technical details:

- Latent dimension ~128
- 8-layer MLP with skip connections
- Training via reconstruction loss on sampled SDF points
- Dataset: ShapeNet + ABC CAD subset

B. Semantic Part Decomposition Module

DeepSDF's latent space isn't disentangled. To support semantic editing, I will introduce a **part decomposition** step:

Two possible approaches:

1. **Graph-based decomposition:**
    - Convert mesh ↔ graph
    - Use a **Graph Neural Network** to identify consistent parts across the dataset
    - Inspired by "Neural Parts: Learning Part-aware Shape Representations" (CVPR 2021)
2. **SDF patch clustering:**
    - Sample local SDF neighbourhoods
    - Use k-means or spectral clustering
    - Extract part-level subspaces

This produces **part-level latent codes** → {z_body, z_handle, ...}

C. Parameter-to-Latent Mapping Network

I will define semantic parameters, e.g.,

- handle_length
- handle_thickness
- body_radius
- curvature
- leg_height

A lightweight MLP will be trained to map these parameters to **latent shifts**:

$$\Delta z = f_{\Theta}(\text{parameters})$$

This is the “edit engine” of the system.

Technical note:
Δz is applied only to the relevant part vector (e.g., z_handle).
Latent regularization ensures changes remain local.

D. Constrained Shape Optimization (Differentiable Constraints)

Editing a neural implicit field can break geometry.
To prevent invalid shapes, I will add differentiable losses:

- **Self-intersection penalty**
- **Surface smoothness (Laplacian regularization)**
- **Volume preservation**
- **Part-boundary continuity loss**

This produces geometrically valid, smooth, manufacturable shapes.

4. Literature Inspirations –

- DeepSDF (Park et al., CVPR 2019)

  Represents 3D shapes using a continuous signed distance field. Provides high-quality reconstruction but latent codes are not editable.

- Occupancy networks (Mescheder et al., CVPR 2019)

  Implicit representation that predicts whether a point is inside an object. Flexible, but again lacks semantic part control.

- Neural Parts (Zhu et al., CVPR 2021)

  Learns consistent part decomposition across shapes. Inspires the “semantic structure learning” portion of our pipeline.

- MeshGraphNets (DeepMind, 2020)

  Graph-based reasoning over meshes. Helps in designing mesh-level constraints during shape editing.

- NeuS and IDR (2021)

  Differentiable rendering techniques that allow shape supervision from rendered images.

- PointFlow / ShapeFlow

  Generative modelling for 3D point clouds using normalizing flows.

5. Dataset Plan

Primary Datasets -

Dataset	Use
ShapeNet v2	Main dataset: mugs, chairs, lamps, tables
ABC CAD Dataset	High-quality mechanical parts for fine-grained editing
PartNet	Category-structured part annotations
3D Warehouse (optional)	Additional real-world meshes

Sample Classes Used:

- Mugs (body + handle)
- Chairs (seat + legs + backrest)
- Lamps (base + pole + shade)
- Tools (blade + handle)

Sampling Strategy:

- Uniform sampling of SDF points
- Near-surface oversampling for accuracy

6. Mid Evaluation Goals (Dec 19, 2025):

By mid-eval, I will deliver **three concrete and measurable outputs**:

A. Shape Prior (SDF Model)

- Trained on a single category (mugs or chairs)

- Reconstructs shapes with low Chamfer distance

B. Part Decomposition Prototype

- Clear part separation on 10–20 shapes
- Visualizations of part-wise clusters

C. One Working Semantic Edit

Example:
Increasing handle length by X% results in a smooth, valid deformation.

Mid-Eval Deliverables:

- 6–10 reconstructed shapes
- Part segmentation visualizations
- Before–after editing examples
- Small progress video

7. End Evaluation Goals (Jan 2, 2026)

By final submission, I aim to deliver:

- Multi-category Shape Editing (2–3 categories):

  Supports mugs + chairs + one more category.

- 3–5 Semantic Edit Controls:

  At least 5 parameters mapped to Δz shifts.

- Full Constrained Optimization:

  No self-intersection, smooth surfaces, etc.

- Interactive UI Demo:

  Gradio/Blender interface with sliders → updated shape.

Full Evaluation Suite:

- Chamfer distance
- Edit correctness (% change in geometry)

- Mesh quality metrics
- Ablation studies

8. Final Documentation + Video

Detailed Timeline –

Date Range	**Tasks**	**Deliverables**
Dec 6–9	- Literature review of DeepSDF, OccNet, Neural Parts - Set up dataset loaders for ShapeNet - Implement basic SDF training pipeline	Initial SDF reconstructions
Dec 10–13	- Train SDF prior on mug category - Evaluate reconstruction quality - Begin part decomposition experiments (GNN or clustering)	Clean reconstruction + early part clusters
Dec 14–18	- Finalize part decomposition model - Implement Δz learning module for one parameter - Test simple edits on mugs	One working edit; mid-eval prototypes
Dec 19 — MID EVAL	Submit: shape prior results + part decomposition + simple edit + video	MID-EVAL SUBMISSION
Dec 20–24	- Add more parameters (handle thickness, body radius) - Implement differentiable constraints (smoothness, intersection)	2–3 working edits with constraints
Dec 25–28	- Extend to another category (chairs) - Multi-part latent control - Build interactive UI	Multi-category edits + UI beta
Dec 29–Jan 1	- Full evaluation suite - Ablations and comparisons - Final polishing + video recording	Final results ready
Jan 2 — FINAL	Submit final PDF, UI demo, results video	FINAL SUBMISSION

9. Tech Stack –

- **PyTorch** — core training
- **PyTorch3D** — mesh operations + differentiable rendering
- **Blender** — visualization
- **ShapeNet / ABC CAD / PartNet** — datasets
- **WandB** — experiment tracking
- **Gradio** — interactive UI

10. Feasability –

Why it is doable in the given time –

- Implicit SDF learning is well-documented
- ShapeNet + ABC datasets are standard
- Part decomposition methods are easy to prototype in Python
- Δz mapping is light and fast to train
- Constrained optimization uses differentiable losses available in PyTorch3D

Expected Challenges -

- Semantic disentanglement may not be perfect
- Some edits might create invalid topology
- Latent spaces may need tuning (locality vs globality)

I will mitigate this with -

- strict latent-space regularization
- hybrid cluster + neural segmentation
- multiple ablations and visual debugging

About Me –

I am a sophomore with strong interest in 3D vision, geometric deep learning, and generative models. Over the next few weeks, I aim to deepen my understanding of:

- implicit neural fields
- mesh processing
- differentiable optimization
- shape analysis
- transformer-based generative models (for later extensions)

This project aligns exactly with my technical interests and long-term goal of working in 3D generative AI and robotics.