Towards Arbitrary Local Editing of Neural 3D Shape Representations

Master Thesis

by

Roel Webb

To obtain the degree of

Master of Science in Robotics
at the Delft University of Technology

To be defended publicly on

Monday October 6, 2025

Studentnumber	4643674
Faculty:	Mechanical Engineering (ME), TU Delft
Department:	Cognitive Robotics (CoR)
Supervisors:	Chirag Raman — Holger Caesar

Examination committee:

Dr. Holger Caesar (Chair) — Intelligent Vehicles, TU Delft
Dr. Chirag Raman — Pattern Recognition and Bioinformatics, TU Delft
Dr. Petr Kellnhofer (External member) — Computer Graphics and Visualization, TU Delft

TU Delft

Towards Arbitrary Local Editing of Neural 3D Shape Representations

Roel Webb
TU Delft
Netherlands

Dr. Chirag Raman
Pattern Recognition and Bioinformatics group
TU Delft
Netherlands

Dr. Holger Caesar
Intelligent Vehicles group
TU Delft
Netherlands

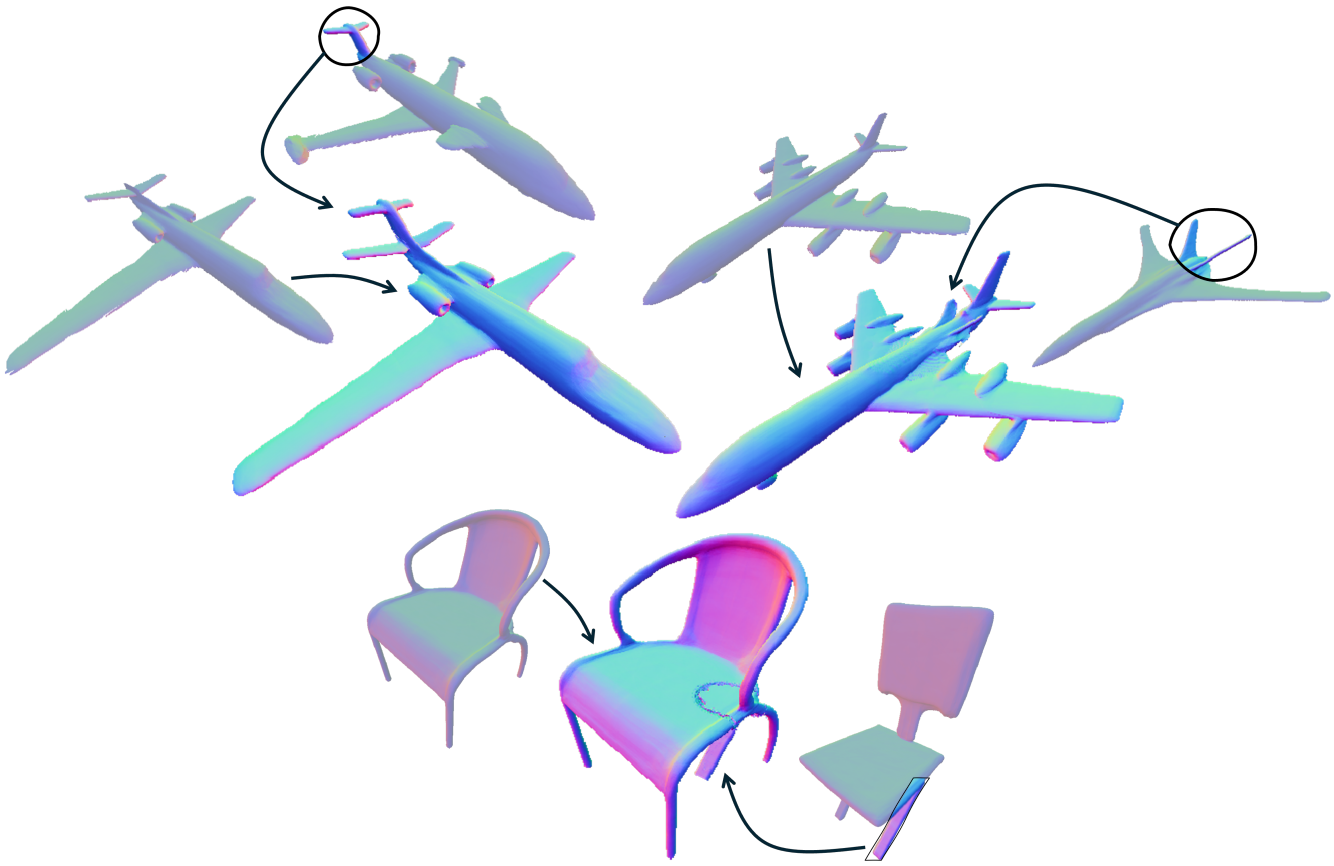


Figure 1: Novel shapes generated with our proposed method

Abstract

The recent explosion in research on 3D generative AI has shown that learning based editing methods can generate a wide range of shapes in an intuitive and low effort manner. However, current methods show limited local editing control due to the coarse shape representations they are based on, as well as shared or propagated geometry information propagating deformations to a larger region. Methods for theoretically unlimited fine editing exist, however they generally lack shape understanding, which is essential for reducing the effort involved in 3D shape editing. We propose a flexible shape representation in combination with an editing procedure to produce theoretically arbitrary fine local edits with learned shape understanding. Our work builds upon the learning based shape representation and generation method SPAGHETTI, which enables meaningful part mixing and interpolation. Our method enables the selection of a local surface region, which can be transferred through mixing or interpolation to a target shape guided by learned semantics. To empirically assess the locality of edits, we propose a new metric, which we use to conduct experiments on two baseline local shape mixing and interpolation methods. Our locality metric and surface displacement visualizations show that our method achieves more localized edits than the baseline methods, which yield significant deformation propagation.

Keywords

local shape editing, 3D shape generation, neural implicit representations

1 Introduction

A wide range of fields including robotics, computer graphics and product design rely on 3D modeling applications to generate 3D models. A common goal for 3D shape generation is to produce high-quality shapes that realize the creator's vision as quickly, accurately and effortlessly as possible [19]. Traditional 3D modeling applications offer precise control, but generally require significant manual effort and expertise. Learning-based 3D generative methods address this by generating shapes that adhere to learned geometric and structural priors, enabling intuitive editing with minimal input [32].

Recent editing methods learn to disentangle and represent shape information, such as local geometry or structure, able to control them within the shape generation pipeline during inference [11], [13], [15]. Such disentangled and controllable shape representations enable novel combinations beyond the training distribution, such as mixing and interpolating local geometry with smooth transitions. However, these methods are limited by the resolution of spatial decomposition in their representations. For example, in part-based models the fineness of editing is limited to the size of decomposed parts and sharing information between parts can cause edits to propagate beyond the intended region, as seen in our experiments (Figure 6, Figure 7). To fully realize a creator's specific vision, these models require a shape representation that encodes shapes as finer local surface regions, enabling finer editing control.

In this work, we present, to the best of our knowledge, the first learning-based feature transfer method whose editing fineness is only limited by the size of intermediate surface geometry of selected donor and base surface, supporting arbitrary small surface selections. Concretely, even the finest details can be transferred while avoiding deformation outside the selected surface regions.

Our method builds on SPAGHETTI [11], which introduces part-aware shape editing, including part mixing and interpolation, through unsupervised part decomposition and part information sharing. In SPAGHETTI, each shape is decomposed and processed by a transformer-based mixing network to produce contextual embeddings Z^c. The occupancy of a query coordinate is predicted with a transformer and MLP based decoder with contextual embeddings as input. By weighting the transformer output on a part level for two shapes, part mixing and interpolation is achieved.

To edit at a finer scale than the decomposed parts, our method introduces a part weight field to control the strength of a donor and base surface feature. Instead of using a single uniform part interpolation weight encoding part interpolation over the full $R3$ query domain, we enable the varying of part weights per query coordinate. As a result, the decoder input at arbitrary scale regions can be selectively controlled, enabling edits at any scale. For example, rather than interpolating a full part, such as a car door with a mirror, our method can localize the interpolation to only the mirror region, leaving the rest of the door unchanged. We introduce two mechanisms for selecting a local donor surface region and specifying transfer strength, together with a blending field update procedure that transfers donor features to the semantically corresponding and spatially localized base shape region.

To evaluate the largely unexplored aspect of edit locality, we introduce a new metric in Section 5 capturing the locality of surface deformations as result of an edit. Using this metric we perform quantitative experiments to compare our proposed method against the two baseline feature transfer methods SPAGHETTI [11] and Neural Wavelet (2024) [13] in section 6 showing that our method performs more local edits than the baseline methods. Additionally we visually compare the regions and propagation of edits between methods by visualizing the surface displacements per surface point.

We list the following contributions: (i) A quantitative edit locality metric for assessing how spatially constrained an edit is in 3D shape manipulation tasks. (ii) A quantitative and qualitative analysis of edit locality on two baseline methods [11], [13], and our proposed method, performing measurements on each method using our proposed metric, and visualizing how edits can propagate deformations. (iii) A new semantically guided local shape editing method, with edit locality limited only by the surface displacement between the transferred feature and the corresponding target region.

2 Related Work

Since neural implicit fields were first introduced to represent 3D shapes [25], [7], [22] a wide range of neural shape representations [29], [26] and shape editing methods [32], [26] have been introduced in the field of 3D generative AI. At their core, editing methods are shape generation methods in which an intuitive input, such as a selected surface point translation or spatial mask selecting a shape region, directly controls the generated shape representation.

2.1 Shape Representations

Early neural implicit field methods generally model the distribution of training shapes with an encoder–decoder framework, where an encoder maps a shape to a latent code and a decoder generates the implicit field representing the corresponding shape. Training to reconstruct a set of shapes results in a manifold of latent codes in latent space which generate valid shapes, with similar shapes embedded close together and latent codes encoding common interpretable shape features [25], [7], [22]. This enabled smooth and meaningful interpolating of latent codes and thus shapes, interpolating between full shapes generating novel shapes not present in the discrete training set. However, these shapes remained constrained to producing variations of known entangled global structures and geometry present in the training data, thus not directly presenting novel editing methods beyond global shape interpolation.

To enable more expressive and controlled manipulation, new flexible, disentangled and localized shape representations were introduced by disentangling parts [9], describing geometry locally [17], [16], [27], separating structure or pose from geometry [9], [23], [24] or separating levels of detail [10], [30], [16].

While several representations disentangle shapes into local regions to support editing, their decomposition fineness remains limited in practice. Part-based methods such as LDIF [9] fix the number of

parts during training, where increasing part count risks breaking semantic consistency and raises memory and computational costs, whereas grid-based methods such as Neural Wavelet (2022) [16] scale cubically in memory and computation with grid resolution.

2.2 Shape Editing Methods

Several shape representations are trained with manually disentangled parameters as input, such as joint angles [24] or pose parameters [1], [23]. Others require less supervision to disentangle features, such as ImFace [33] able to disentangle facial expression and identity information for each face only requiring a label indicating a face has a neutral expression during training. Shape editing then simplifies to adjusting the input parameters or interpolating features partly disentangled before training, however this requires costly manual labeling.

Several editing methods apply traditionally editing concepts to neural implicit fields, deforming a shape by translating the query coordinate input, such as cage-based deformation [31], linear blend skinning [23] or ARAP based editing [2].
To intuitively control the global latent code representation of a shape DualSDF [10] uses a coarse shape representation defined by spheres which can be intuitively scaled and transformed. By connecting the coarse representation with the detailed representation, the fine shape can be adjusted through simple sphere adjustments.
To incorporate semantics in the global shape editing process, DIF-Net [8] generates shapes by learning to translate a queried point to its semantically corresponding location on a shared template field, which learns all possible structures, combined with a correction to the SDF output of the template field to enable and disable structures.

SPAGHETTI [11] builds on the LDIF [9] shape representation which decomposes a shape into 16 parts and disentangles structural information (part position, orientation and scale) from normalized part geometry. By sharing information between parts after this part decomposition and structure-geometry disentanglement, SPAGHETTI can plausibly mix and interpolate parts, generating novel part combinations as well as translating, rotating and scaling the shape at the learned part level. Both the part decomposition and structure-geometry disentanglement enable generation of shapes not present in the global shape distribution of the training data. The locality of edits remains limited by the size of decomposed parts as well as part edits propagating due to part information sharing.
To perform more fine-grained editing, Neural Wavelet (2024) [13] proposes a new local shape mixing and interpolating method on the grid-based representation Neural Wavelet (2022) [16] which encodes local geometry as wavelets encoded with 46 by 46 by 46 coarse wavelet coefficients. After selecting local regions of two shapes to keep or replace, the diffusion based mixing and interpolation procedure generates new wavelet coefficients incorporating the local regions to keep. The locality of the edit is limited by the resolution of the coarse coefficient grid and scale of represented wavelets with additional deformation propagation due to the denoising process. Building further upon Neural Wavelet [16], [13], CNS-Edit [15] introduces more edit operations on a single shape as well as a new semantically aware coupled neural shape representation for more intuitive and deeper shape understanding, but lacks mixing and interpolation between multiple shapes.

Several neural editing methods do not rely on the modeling of shape distributions to guide the shape editing process and solely optimize latent or network parameters to a geometric input such as a surface displacement [3], [28]. Without learning from a training set of valid shape samples, these methods lack meaningful shape understanding lacking the guidance to reduce required manual effort and expertise. However, these methods present theoretically unlimited editing fineness by adding more sampling points during edits capturing higher resolution details, as well as giving higher expressiveness due to the geometric edits not being limited to the features seen in a finite training set.
As a result, existing methods either offer intuitive, learning-based editing with limited locality or fine geometric control without learning-based guidance.

2.3 Edit Metrics

The performance of a 3D shape editing method depends on its application as well as having various aspects such as expressiveness, quality and control each requiring its own metrics.
To measure the ability to generate diverse shapes without collapsing to a small subset of modes, the coverage of a training set distribution is typically measured using Coverage (Cov), Minimum Matching Distance (MMD), 1-Nearest Neighbor Accuracy (1-NNA) [21] as well as Edge Count Difference (ECD) [17], which all rely on widely known similarity scores such as Chamfer Distance (CD) and Earth Mover's Distance (EMD).

Currently there is no widely used metric to represent the ability to generate novel shapes further away from the training data distribution. However, while having high coverage, Neural wavelet (2022) [16] report the LFD [6] similarity score between generated shapes and the nearest training data sample. With high LFD scores indicating shapes being further from the training set, higher novelty can be indicated, as was visually confirmed with novel but plausible shapes combining plausible local structures and geometries.
To asses the (visual) quality of edited shapes, commonly rendered images from the shape are assessed on metric such as Fréchet Inception Distance (FID) [12], shading-based S-FID [34] and Kernel Inception Distance (KID) [4]. Additionally CNS-Edit [15] performs a survey with the Quality Score (QS) metric reflecting visual appeal proposed by Hu et al. [14].

Control is a critical yet underexplored aspect of shape editing. CNS-Edit [15] utilizes the Matching Score (MS) [14], which measures how well edits match user intent. While this provides valuable insight into a user's perceived control, it is subjective and lacks quantitative precision.
An essential part of control is spatial control: the ability to restrict edits to specific regions without unintended changes elsewhere. Despite its importance, no widely used metric exists to assess spatial precision in neural shape editing. To address this gap, we propose

an edit locality metric, providing the first quantitative measure of spatial control in 3D shape editing.

3 Problem Definition

We define a watertight shape as a surface that separates inside and outside regions in $\mathbb{R}^3$. The surface is defined by the set of points $S = \{\mathbf{p} \in \mathbb{R}^3 \mid f(\mathbf{p}) = t\}$, where $f : \mathbb{R}^3 \to \mathbb{R}$ is an implicit field assigning a scalar value s to each 3D query point $\mathbf{p}$. Here s represents the probability of being inside the shape (occupancy) or the signed distance to the closest surface (negative inside and positive outside the shape). The iso-value t defines which associated points lay on the surface, with $t = 0$ for signed distance fields and $t = 0.5$ for occupancy fields. The inside and outside regions are given by $\Omega_{\text{in}} = \{\mathbf{p} \in \mathbb{R}^3 \mid f(\mathbf{p}) < t\}$ and $\Omega_{\text{out}} = \{\mathbf{p} \in \mathbb{R}^3 \mid f(\mathbf{p}) > t\}$. Modifying the output values of $f(\mathbf{p})$ near the surface induces local deformations and thus defines shape edits.

3.1 Local Shape Editing

We define local shape editing as replacing values of a base implicit field $f_b(\mathbf{p}_b)$ with values of an insert implicit field $f_i(\mathbf{p}_i)$ within a bounded spatial region V_r leaving the surface outside this region unaffected, as shown in Figure 2. As a result the base (original) surface $S_b = \{\mathbf{p} \in \mathbb{R}^3 \mid f_b(\mathbf{p}_b) = t\}$ is locally deformed. The coordinate system of f_b is the global frame F_b and f_i is defined in its own frame F_i, with query points $\mathbf{p}_b$ and $\mathbf{p}_i$ expressed in their respective frames.

The edit region is specified by a bounding volume

$$V_r = \{\mathbf{p} \in \mathbb{R}^3 \mid g(\mathbf{p}_r) = 1\}, \tag{1}$$

where $g : \mathbb{R}^3 \to [0, 1]$ is a scalar indicator function in the replacement frame F_r and $\mathbf{p}_r = T_{br}\mathbf{p}_b$ is the transformed query point from the base to replacement field. By aligning the replacement frame F_r and F_i, the edited implicit field is defined as:

$$f_e(\mathbf{p}_b) = \begin{cases} f_i(T_{br}\mathbf{p}_b), & \mathbf{p}_b \in V_r \\ f_b(\mathbf{p}_b), & \mathbf{p}_b \notin V_r \end{cases} \tag{2}$$

To guarantee that f_e defines a continuous watertight surface, discontinuities are not allowed, therefore the base and insert surfaces must coincide along the boundary of V_r, defining a shared edge E:

$$E = \{\delta V_r \cap S_b\} = \{\delta V_r \cap S_i\} \tag{3}$$

where $S_i = \{\mathbf{p} \in \mathbb{R}^3 \mid f_i(\mathbf{p}_i) = t\}$. Additionally, inside/outside orientation must be preserved at the shared edge E.

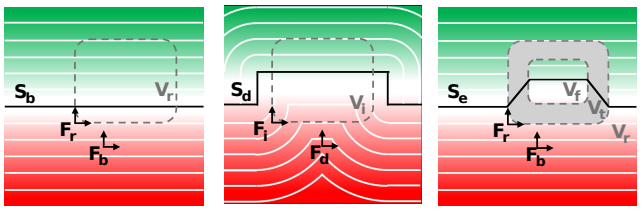
$$\langle \nabla f_b(\mathbf{p}_b), \nabla f_i(T_{br}\mathbf{p}_b) \rangle > 0, \quad \mathbf{p}_b \in E. \tag{4}$$

When no frame is indicated and subscripts are omitted, query points are expressed in the global base frame F_b.

3.2 Local Shape Mixing

Local shape mixing combines local surfaces from different shapes into a single new shape. It can be expressed as a special case of local shape editing where the insert field $f_i(\mathbf{p}_i)$ is defined as a local region of a donor field $f_d(\mathbf{p}_d)$ representing a local region of a donor surface $S_d = \{\mathbf{p} \in \mathbb{R}^3 \mid f_d(\mathbf{p}_d) = t\}$ as visualized in Figure 2b.

Formally, $f_i(\mathbf{p}_i) = f_d(T_{id}T_{br}\mathbf{p}_b)$, where $\mathbf{p}_b \in V_i$, F_i is a frame defined relative to the donor frame F_d, and T_{id} transforms coordinates from F_i into F_d. By adjusting F_i, different local features of the donor surface S_d can be selected for insertion into the base.



(a) Base implicit field (b) Donor implicit field (c) Edited implicit field

Figure 2: Local shape mixing by inserting a donor patch into a base shape within a bounded region. A transition volume ensures smooth blending between base and donor surfaces.

Enforcing the edge constraint from Equation (3) restricts valid edits to cases where $\delta V_r \cap S_i = \delta V_r \cap S_b$. In practice, many donor shapes do not contain a surface patch that exactly satisfies this condition. To increase flexibility, we introduce a transition region V_t and transition field $f_t(\mathbf{p}_i)$ defined in F_i that smoothly interpolates between the base and donor surfaces inside the bounding volume as shown in fig. 2c. The combined field is then described by:

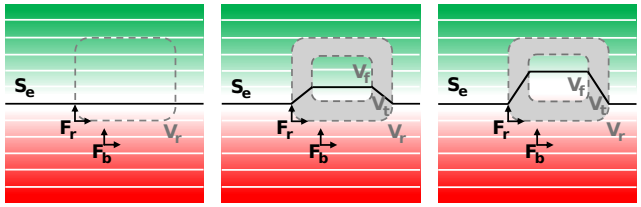
$$f_e(\mathbf{p}_b) = \begin{cases} f_d(T_{id}T_{br}\mathbf{p}_b), & \mathbf{p}_b \in V_f, \\ f_t(T_{br}\mathbf{p}_b), & \mathbf{p}_b \in V_t, \\ f_b(\mathbf{p}_b), & \mathbf{p}_b \notin V_r, \end{cases} \tag{5}$$

where $V_r = V_f \cup V_t$, $\delta V_f \cap \delta V_r = \emptyset$ and $V_t = V_r \setminus V_f$.

To ensure a watertight shape the transition surface S_t inside V_t must satisfy analogous edge and orientation constraint to Equation (3) and Equation (4):

$$\begin{aligned} E_{tb} &= \{\delta V_r \cap S_t\} = \{\delta V_r \cap S_b\} \\ E_{td} &= \{\delta V_f \cap S_t\} = \{\delta V_f \cap S_d\} \end{aligned} \tag{6}$$

$$\begin{aligned} \langle \nabla f_t(T_{br}\mathbf{p}_b)), \nabla f_b(\mathbf{p}_b) \rangle > 0, \quad \mathbf{p}_b \in E_{tb} \\ \langle \nabla f_t(T_{br}\mathbf{p}_b), \nabla f_d(T_{id}T_{br}\mathbf{p}_b) \rangle > 0, \quad \mathbf{p}_b \in E_{td} \end{aligned} \tag{7}$$



(a) Base implicit field with $w = 0$ (b) Edited implicit field with $w = 0.5$ (c) Edited implicit field with $w = 1$

Figure 3: Local shape interpolation: smooth interpolation between base and donor surfaces within a bounded region, enabling continuous morphing of local features.

3.3 Local Shape Interpolation

Local shape interpolation aims to generate intermediate surfaces that smoothly deform between a base surface S_b and a donor surface S_d within a bounded region. We build on the formulation of local

shape mixing, replacing the field inside V_f with the interpolated surfaces given by $h(f_b, f_d, w, \mathbf{p}_b)$, where $\mathbf{p}_b \in V_r$ and $w \in [0,1]$ controls the interpolation between base and donor features as shown in Figure 3. The function $h(f_b, f_d, w, \mathbf{p}_b)$ defines the interpolation behavior, with boundary conditions,

$$\begin{aligned} h(f_b, f_d, 0, \mathbf{p}_b) &= f_b(\mathbf{p}_b), \\ h(f_b, f_d, 1, \mathbf{p}_b) &= f_d(T_{id} T_{br} \mathbf{p}_b). \end{aligned} \tag{8}$$

A simple choice for h is linear interpolation, $h(f_b, f_d, w, \mathbf{p}_b) = (1 - w_f) * f_b(\mathbf{p}_b) + w * f_d(T_{id} T_{br} \mathbf{p}_b)$, however more complex functions can be utilized to enable more meaningful interpolation.
The resulting locally interpolated field can be written as

$$f_e(w_f, \mathbf{p}_b) = \begin{cases} h(f_b, f_d, w_f, \mathbf{p}_b), & \mathbf{p}_b \in V_f, \\ f_t(T_{br} \mathbf{p}_b), & \mathbf{p}_b \in V_t, \\ f_b(\mathbf{p}_b), & \mathbf{p}_b \notin V_r, \end{cases} \tag{9}$$

where w_f defines the interpolation strength.
Due to the boundary conditions on h, setting $w_f = 0$ recovers the original base field, while setting $w_f = 1$ corresponds to local shape mixing. Note that the interpolation function h can also act as the transition field f_t by letting the weight w vary smoothly from 0 at δV_r to w_f at δV_f within the transition region V_t.

3.4 Semantically Plausible Interpolation

Various shape categories, such as animal species or cars, have distinct meaningful features and share common structures. To interpolate plausibly between such shapes, surface points must deform in a way that preserves their semantic meaning to avoid artifacts such as duplicated features (e.g., two noses) or misaligned deformations (e.g., a forehead morphing into a mouth). Transferring features to arbitrary locations is more difficult because such cases are sparsely represented in training data and are outside our scope. Therefore, we assume the selected base and donor regions are approximately aligned, reducing the transforms T_{br} and T_{bd} to identity.
Any smooth shape distribution modeling function constraint by Equation (8) can then serve as full edit field, as w transitions smoothly from 0 on δV_r to w_f on δV_f:

$$f_e(\mathbf{p}) = \begin{cases} h(f_b, f_d, w_f, \mathbf{p}), & \mathbf{p} \in V_f, \\ h(f_b, f_d, w(\mathbf{p}), \mathbf{p}), & \mathbf{p} \in V_t, \\ h(f_b, f_d, 0, \mathbf{p}), & \mathbf{p} \notin V_r, \end{cases} \tag{10}$$

where $w : \mathbb{R}^3 \to [0,1]$ is a spatially varying weight field defined as part of the editing process. With the alignment assumption, intermediate features lie along the modeled shape distribution (e.g., smoothly morphing an airplane nose from cargo to fighter), while $w(\mathbf{p})$ enables fine-grained local control over the influence of base and donor features.

4 Feature-Size Local Editing

Overview. Our method builds on SPAGHETTI [11], which decomposes shapes into part embeddings contextualized by a transformer mixing network and decoded into an implicit field. While SPAGHETTI enables part mixing and interpolation at the level of whole parts, it lacks finer control because part weights are defined globally.
We introduce a controllable continuous weight field represented by *blending anchors* for localized editing. Each anchor is defined by a surface coordinate $\mathbf{p}_i$, a donor–base blending weight $\alpha_i \in [0,1]$, and a semantic coordinate $\mathbf{s}_i \in R3$. Anchor's weights are spatially interpolated across anchors to construct a continuous field $\alpha(\mathbf{p})$, and by updating or introducing new anchors at arbitrary resolution, edits can be performed at fine scales. The semantics $\mathbf{s}_i$ is used to help preserve the semantic meaning of deformed surfaces and mitigate artifacts described in Section 3.
Our pipeline consists of four stages: (1) initialize base anchors and assign donor influence weights to selected surface points, (2) calculate blending weights for donor-selected surface points, (3) compute surface coordinates of the new blending anchors, (4) remove redundant base anchors and combine the final set of anchors.

4.1 Shapes as Blending Anchors

We extend the part-level interpolation framework of SPAGHETTI [11], which represents a shape using two sets of contextual part embeddings, denoted as Z_b^c and Z_d^c for the base and donor shape respectively (see Section 3). In the original formulation (section 4.5 of the paper), interpolation is controlled by a single global weight α, with α specifying the influence of the donor and $1 - \alpha$ that of the base.
To enable spatially varying influence, we replace this global interpolation weight with a continuous field $\alpha(\mathbf{p})$, analogous to the blending field in NeuForm [20] applied on SPAGHETTI. Our field is parameterized by *blending anchors*: an arbitrary number of updatable coordinates $\mathbf{p}_i$, initialized on or updated to a surface. Each anchor is associated with a scalar blending weight $\alpha_i \in [0,1]$, indicating donor–base influence, and a semantic coordinate $\mathbf{s}_i \in \mathbb{R}^3$ that encodes its semantic meaning. These semantic coordinates are later used for semantically guided feature transfer from donor to base.

The continuous field $\alpha(\mathbf{p})$ is obtained by distance-weighting the k nearest anchor weights:

$$w_i = \frac{1 - \frac{d_i}{D_{\text{knn}}}}{\sum_{j=1}^{k} \left(1 - \frac{d_j}{D_{\text{knn}}}\right)}, \tag{11}$$

where d_i is the distance from a query point $\mathbf{p}$ to its i-th nearest anchor and D_{knn} is the distance to the furthest of the k neighbors. The contextual embeddings Z_b^c and Z_d^c can then be blended spatially according to these weights.
Our proposed blending anchors are flexible, since they can be updated, removed, or new anchors can be generated, enabling arbitrarily dense placement of locally varying weights. Higher anchor densities can be used in regions requiring fine-grained blending, while large homogeneous regions with similar donor–base influence require fewer anchors.

Finally, the semantic coordinates $\mathbf{s}_i$ are defined on a shared surface template encoding semantic meaning, making it possible to find correspondences between base and donor surface coordinates. This ensures that anchors not only control spatial blending but also guide the transfer of semantically consistent features.

4.2 Anchor Initialization

We first initialize blending anchors to represent the base shape already having the contextual embeddings for the base and donor. To initialize these anchors, we reconstruct the base shape using marching cubes and extract the mesh vertices, providing an approximately uniform set of surface points that serve as the initial anchor positions. Since the donor features have not been transferred yet, the blending weight of each anchor is initialized to 0, effectively representing a normal base shape. Although the initial base anchors are initialized with donor weight 0, subsequent edits generate anchors with non-trivial blending weights. This enables iterative editing, where edited anchors incorporating donor features can serve as the base for further localized feature transfer.

Once the base and donor shapes are selected, the template shape is computed with global $\alpha = 0.5$ according to SPAGHETTI's [11] proposed mixing method, equally mixing the transformer decoder output of the base and donor shape. To assign semantic coordinates to base or donor surface points, we approximate their correspondence to the shared template surface during interpolation. Specifically, we track how surface points translate when interpolating from either the base or donor toward the template shape. This translation is computed iteratively using the principle of boundary sensitivity introduced by Berzins et al. [3]. We adapt the *boundary equation 2* of Berzins et al. [3], by replacing network parameters Θ, with the blending weights input α, as well as disregarding the tangential components δx_t, to write the displacement of a surface point under finite parameter update $\Delta\alpha$:

$$x_{i+1} = x_i + \frac{-\nabla_{x_i} f \, \nabla_\alpha f^\top \, \Delta\alpha}{\|\nabla_{x_i} f\|^2}, \tag{12}$$

Here x_i denotes the query point input to the network starting at x_0 on the surface and x_n reaching the template, where α is updated from the base or donor (0 or 1) to the template (0.5) in n steps, with the fraction $\Delta\alpha$ equal to $0.5/n$. We perform this procedure for $n = 25$ steps, gradually translating base and donor surface points along their normal toward the template surface.
While this approach generally works well for roughly aligned shapes, neglecting tangential components can occasionally yield incorrect correspondences, resulting in semantic mismatches and artifacts. We further discuss these cases in the limitations and future work section.

4.3 Donor Anchor Sampling

To generate donor blending anchors, we again assign the base and donor contextual embeddings. The donor anchors are initialized from surface points sampled within a selected donor region, and each is assigned a blending weight that controls its influence relative to the base. We propose two sampling strategies to specify the donor region and weights:

(i) 3D spherical region. A surface coordinate is chosen as the edit center, and a maximum radius is defined to bound the donor region. The occupancy field or (T)SDF is sampled on a grid covering the edit volume with sufficient density to capture local details. Using marching cubes, we extract the surface mesh and select vertices within the radius. Each selected point is assigned a donor influence weight based on its normalized 3D distance to the edit center, using the template

$$(1 - r^4)\ \exp(-r^2),$$

where r is the normalized distance clamped to $[0, 1]$. This assigns 1 at the edit center and decays smoothly to 0 at the radius boundary, producing a continuous donor–base transition.

(ii) Camera-based sampling. Alternatively, a grid of rays is raymarched from a virtual camera to find donor surface points. The projected coordinates are weighted by their radial distance to the image center using the same template

$$(1 - \|\mathbf{q}\|^4)\ \exp(-\|\mathbf{q}\|^2),$$

where $\mathbf{q}$ are the normalized 2D coordinates, clamped to $[0, 1]$. This assigns 1 at the image center and smoothly decays to 0 at the edges, yielding an intuitive view-based selection consistent with the 3D case. The effective size of the selection region is controlled by the focal lengths f_x and f_y of the virtual camera, with larger values yielding a more localized selection and smaller values producing a broader region.

In both cases, the sampled donor surface points serve as the donor blending anchors, initialized with their respective weights. To ensure semantic consistency, we iteratively displace these donor anchors toward the shared template surface using the boundary sensitivity procedure described in Section 4.2. This provides each donor anchor with a semantic coordinate on the template, enabling alignment with the corresponding base anchors.

4.4 Blending Weight Computation

We compute updated donor-anchor blending weights in two stages. First, we treat the base anchors' weights as a continuous field over semantic space and query this field at each donor anchor's semantic coordinate. Concretely, for a donor anchor with semantic coordinate $\mathbf{s}_i^{(d)}$, we find k nearest neighbors base anchors in semantic space and interpolate their weights using linear distance weighting (as in Section 4.1). This yields the semantically corresponding base weight for each donor anchor.

Second, we update the donor anchor weights to reflect the new donor influence corresponding to each anchor's semantics, essentially these weights contain the final edit blending weights. The updated donor weight incorporates the corresponding base weight by scaling it and adding the donor weight:

$$\alpha_i^{\text{edit}} = (1 - \alpha_i^{\text{donor}})\, \alpha_i^{\text{base}} + \alpha_i^{\text{donor}}, \tag{13}$$

where α_i^{donor} is the donor weight and α_i^{base} is the corresponding base weight. This formulation ensures a smooth, semantically consistent transition: higher α_i^{donor} increases donor influence locally, while regions with low α_i^{donor}.

4.5 Surface Coordinate Update

Given the final donor anchor weights, we update donor anchors by iteratively displacing their surface coordinates so that the anchor coordinates, and thus the blending field, match the user-intended edited surface. Starting from donor-surface locations, we perform

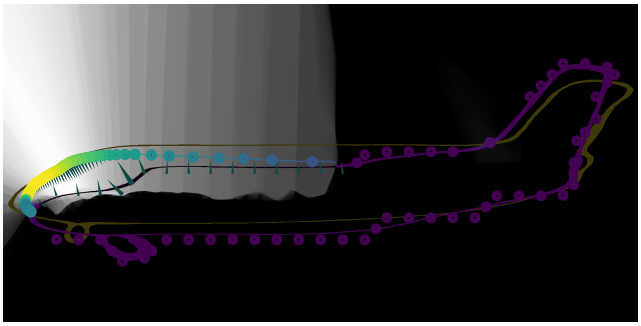


Figure 4: Visualization of a continuous blending field, ranging from full base influence (black) to full donor influence (white), created with blending anchors visualized by their surface coordinates plotted as colored circles; the edited surface passes through these anchors. The base shape (dark purple) and donor shape (dark yellow) are shown behind the edited shape. Dark green arrows show the projection from translated base surface points and their corresponding edit shape points to their locations on the template shape. Note that the visualized blending anchors are sparser than those used in the experiments and 3D visualizations.

boundary-sensitive normal displacements under the changing interpolation weights, exactly as in Section 4.2. Concretely, we step the interpolation weights from the donor anchors toward the final edit weights, and at each step move points according to the boundary sensitivity update in Equation (12). The goal is to locate, for each anchor, the surface point that preserves its original semantic meaning while following the trajectory induced by interpolating from donor to intermediate shapes represented by the blending weight. This yields updated donor anchor coordinates that remain semantically consistent while aligning with the edited surface.

4.6 Anchor Pruning and Merging

With the donor anchors finalized the base and donor anchors can be combined. Since base anchors are still present near the updated donor anchors as well as having similar semantic meaning, we remove base anchors near donor anchors for both the surface coordinates as well as the semantics. Either the grid size of marching cube extracted donor surface points or the largest distance between donor coordinates in any dimension is used as approximate measure to remove base anchors by their distance with a safety factor. After the removal of overlapping base anchors both base and donor anchor sets can be combined to represent the new edited shape with its contextual embeddings and continuous blending field as input for the SPAGHETTI [11] decoder. A continuous blending field generated from blending anchors for two shapes is shown in fig. 4, together with the blending anchors, the base and donor surfaces, and their corresponding projections onto the template shape.

5 Edit Locality Metric

While full shape dissimilarity metrics such as Chamfer distance and Earth Mover's Distance are widely used, there is, to the best of our knowledge, no widely adopted metric that directly quantifies the locality of edits. By locality, we mean the spatial region where a shape has changed as result of an edit. To address this, we propose a new scalar metric that captures the size of the three-dimensional region modified between a base and an edited shape, enabling direct quantitative comparisons of edit locality across methods.

Given two shapes (base and edit), both normalized to fit inside

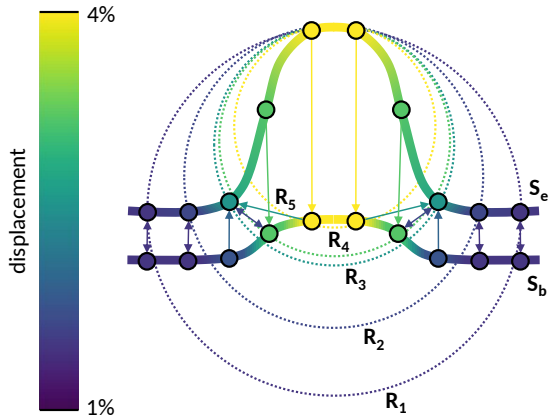


Figure 5: Illustration of the proposed edit locality metric. Arrows denote nearest neighbor distances between corresponding surface points on the base (S_b) and edited (S_e) surface. Each point is colored by the maximum nearest neighbor displacement assigned from both directions, normalized as a percentage of a unit sphere radius. The dashed circles (R_1–R_5) show minimal bounding spheres at increasing displacement thresholds.

a unit sphere, we identify the edited region as the set of surface regions that differ between the base and edited shape. For simplicity and interpretability, we define edit locality as the radius of the minimal bounding sphere that encapsulates all significantly displaced surface points. This value is expressed as a percentage of the unit sphere's radius, providing a normalized, single-scalar measure of locality.

To identify displaced surface points, we reconstruct surface meshes for both the base and edited shapes using marching cubes on identical grids, so that mesh vertices approximate the surface points. For every surface vertex in both shapes, we compute its nearest neighbor distance to the other shape's surface. To robustly capture cases where, for example, a large spike appears on one shape and only a small bump exists on the other, we assign to both points in each nearest-neighbor pair the maximum displacement observed in either direction. This ensures that local deformations are not underestimated when nearest neighbor correspondences are not one-to-one, as visualized in Figure 5.

Vertices with a displacement exceeding a set threshold are considered to be displaced, and the collection of all such points defines the dissimilar surface region. Thresholding is used to ignore negligible surface displacements, which could otherwise result in overestimation of the edited region due to minor, non-perceptible changes. We therefore report the bounding sphere radius as a function of varying

displacement thresholds, allowing for a more nuanced assessment of edit locality.

6 Experiments

Our experiments demonstrate that our method produces significantly more localized edits than SPAGHETTI [11] and Neural Wavelet (2024) [13]. We report results both quantitatively, using the proposed locality metric in Section 5, and qualitatively, through surface displacement visualizations and generated examples. For both experiments, our method achieves stable, spatially confined edits, whereas baseline methods show substantial propagation beyond the intended region.

6.1 Results

6.1.1 Locality metric. The results of our locality metric (Figure 6 and Table 1) highlight key differences in the spatial extent of edits produced by each method. For Neural Wavelet (2024) [13], bounding sphere radii are surprisingly large at the lowest displacement thresholds (<3%), indicating that small but widespread deformations occur throughout the shape, even when visual changes are minimal. This reflects the denoising and harmonization characteristic of diffusion-based models: minor surface changes are distributed globally, although these are typically not perceptible. As the displacement threshold increases, the bounding sphere radius for Neural Wavelet rapidly decreases, consistent with its strict masking procedure and highly local part mixing at larger scales; the low interquartile range (IQR) here indicates high consistency across edits.

SPAGHETTI [11], in contrast, produces relatively large bounding sphere radii for most thresholds. This is a direct consequence of its coarse part decomposition (16 parts per shape) and the transformer-based mixing network, which propagates the influence of an edit beyond the immediate region of the selected part. Displacement maps confirm that edits extend into adjacent structural regions rather than being perfectly confined. The moderate IQR values reflect variability in part sizes and spatial extent of the edits.

Our proposed method shows stable and consistent locality measurements. Because the edit radius mask scales with the donor–base displacement and the metric assigns the largest bidirectional displacement, the measured radius closely matches the intended edit region. Crucially, the radius does not shoot up at small inclusion thresholds, indicating that minor deformations remain confined rather than propagating widely across the shape.

Overall, these results confirm that our method reliably enforces edit locality, while Neural Wavelet and SPAGHETTI both suffer from edit propagation, Neural Wavelet through widespread small deformations and SPAGHETTI through coarse edits that affect large shape regions. The proposed metric therefore provides a clear quantitative assessment of both the scale and the consistency of edits.

6.1.2 Displacement maps. The displacement maps in Figure 7 show significant differences in the spatial extent and magnitude of surface deformations between editing methods, with relatively comparable behavior per method for all shape pairs.

For SPAGHETTI [11], the intended edit for each base-donor pair is not always clearly defined, although in the second row the selected chair leg is distinctly edited. Although the method appears to transfer the donor leg features onto the base, it also introduces significant deformations in unrelated areas, including the armrest, the opposite leg, and a large portion of the seat. This demonstrates that local part edits in SPAGHETTI can result in the propagation of changes beyond the intended region, affecting additional and potentially unwanted areas of the shape.

Neural Wavelet (2024) [13] shows only very small displacements, as also shown in the quantitative results in Section 6.3.1. In the displacement maps, most changes remain below the 1% threshold and are difficult to detect by the measurements. Displacements slightly above 1% are rare and small, however they are still widely distributed across certain shapes instead of being localized. As discussed in Section 6.3.1, this behavior is likely due to the (final) denoising step combined with the small region being mixed, causing the edit to dissipate, resulting in widespread but minimal surface changes.

In contrast to the baseline methods, our proposed method produces more localized displacement, with nonzero values limited to the intended edit region. The boundaries of the edit are sharply defined, resembling the edit radius described in Section 6.3.1. For the majority of the shape surface, no displacements above 1% are detected, while within the edit region, small displacement are observed.

The displacement map visualizations qualitatively support the quantitative findings, showing that our method enables varying strength local editing with minimal unintended deformation spreading to regions away from the intended edit region.

6.2 Results of Local Feature Transfer

Several resulting edits from our local semantic feature transfer method are illustrated in Figure 1. The generated airplanes have been generated by sampling the donor selection coordinates with a virtual camera as well as using the projection method described in section 4.3. The chair has been generated with the 3D spherical region method discussed in Section 4.3.

6.3 Experiments Setup

We evaluate the spatial locality of shape edits produced by our method, SPAGHETTI [11], and Neural Wavelet (2024) [13], using both our proposed quantitative locality metric (Section 5) and qualitative displacement maps that visualize the strength of surface point displacements. These baselines were chosen for their ability to transfer local features between shapes through mixing and interpolation, while methods such as DIF-Net [8] and DualSDF [10] only support global shape editing and more recent CNS-Edit [15] and 3DNS [28] are unable to blend different shapes. For fair evaluation, we use pretrained models for all methods and restrict our evaluation to the ShapeNet [5] chairs category, as pretrained models for both SPAGHETTI and Neural Wavelet are only available for this class. Since the released Neural Wavelet code supports only shape inversion, we reimplemented the shape mixing procedure following the description provided in the paper and supplementary material.

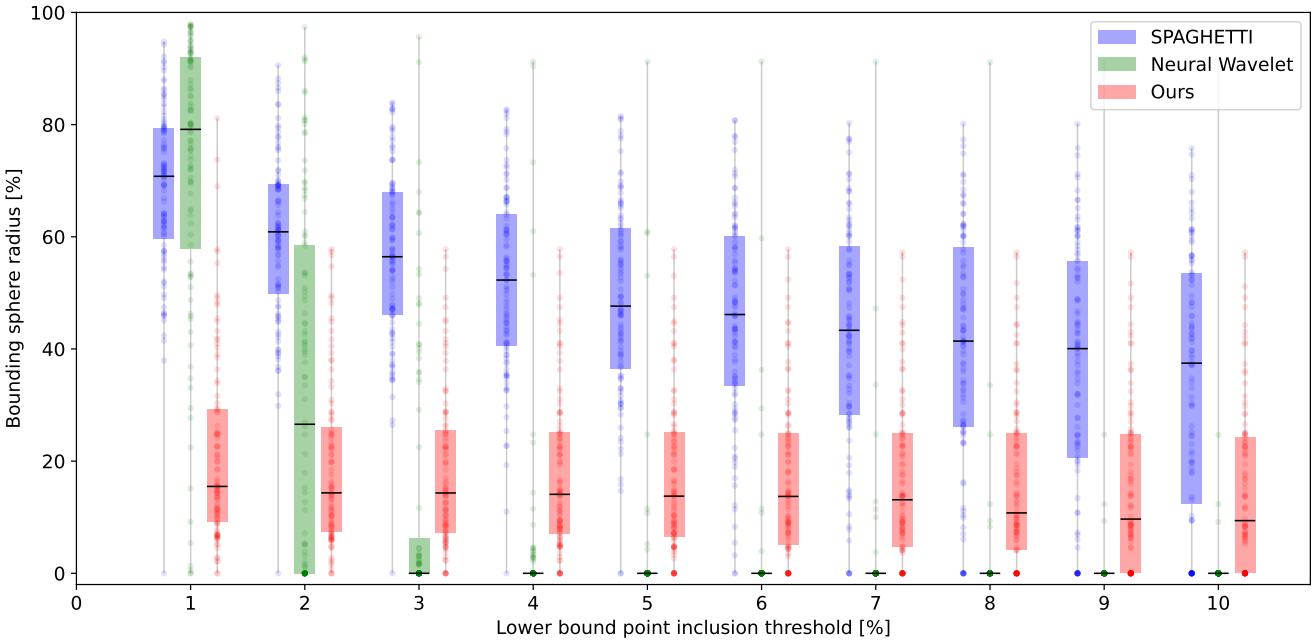


Figure 6: For each of the 100 edits, we flag base- and edit-shape vertices whose assigned displacement d (%) exceeds a threshold t (%), both normalized to the unit-sphere diameter. We apply ten thresholds, from low (flagging nearly all vertices) to high (flagging only larger displacements). At each threshold, we compute the radius of the minimal enclosing sphere of the flagged vertices as a percentage of the unit-sphere radius. The plot shows the median and the central 50% range (interquartile range) of these radii across all edits for each method.

Method	**Bounding sphere radius: Median (IQR)**									
	Lower bound inclusion threshold									
	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
SPAGHETTI	70.8 (19.8)	60.9 (19.5)	56.4 (21.9)	52.3 (23.4)	47.6 (25.0)	46.1 (26.6)	43.3 (30.0)	41.4 (32.1)	40.1 (35.0)	37.5 (41.0)
Neural Wavelet	79.2 (34.2)	26.6 (58.6)	**0.0 (6.3)**	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**	**0.0 (0.0)**
Outs	**15.5 (20.0)**	**14.3 (18.6)**	14.3 (18.3)	14.1 (18.1)	13.8 (18.6)	13.7 (19.9)	13.1 (20.3)	10.8 (20.8)	9.7 (24.8)	9.4 (24.2)

Table 1: Median and interquartile range (IQR) of minimal bounding sphere radii (% of unit-sphere radius) for each method, evaluated across 100 edits. Values are reported at displacement thresholds ranging from 1% to 10%. Lower values indicate more local edits.

6.3.1 Quantitative Edit Locality Measurements. We quantitatively assess edit locality by applying the minimal local edit possible with each method on a set of 100 shape pairs from ShapeNet [5] chairs. Shape pairs are constructed by shuffling all 6,755 chair indices, assigning the first 100 as base shapes and the next 100 as donors, then pairing base i with donor i, ensuring no shape is used as both base and donor. The same pairs are used for all methods for comparability.

For each shape, we use the pretrained shape embedding z_a provided by the SPAGHETTI [11] model. The contextual part embeddings Z_c are deterministically computed by forwarding z_a through the model's decomposition and mixing networks, following SPAGHETTI's inference pipeline. The Z_c embeddings are used as input for both SPAGHETTI and our proposed method during measurements, and to generate meshes for shape inversion with Neural Wavelet (2024) [13]. This shape inversion on Neural Wavelet results in refined latent codes z_{sem}, which are input for the part mixing method in Neural Wavelet. As a result all experiments are evaluated on the same set of shape pairs.

SPAGHETTI: For each base-donor pair, we randomly select one of the 16 part indices (corresponding to the model's learned semantic parts). Editing is performed following the procedure described in the SPAGHETTI paper, where the selected part's contextual embedding in the base shape is replaced with that from the donor. The

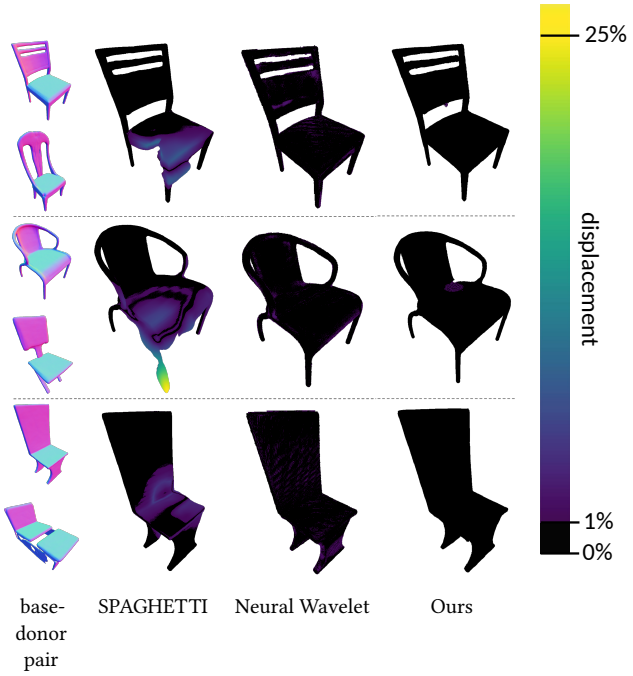


Figure 7: Surface displacement maps (middle three columns) for edited shapes across three selected base-donor pairs. For each pair (left column, with base shape on top and donor below), local surface displacement is visualized using a color gradient, where colors indicate the magnitude of displacement relative to the unit sphere radius. Black denotes displacements below 1%, while values above 25% are colored yellow to improve the perceptibility of differences at lower magnitudes. The visualizations demonstrate that our method results in significantly less edit propagation compared to the baseline methods.

resulting contextual embedding is input to the occupancy network to generate the occupancy field.

Neural Wavelet: For each pair, we randomly select a donor surface coordinate and map it to the corresponding cell in the 46^3 coarse coefficient grid. The mask is then expanded to include all cells within a Manhattan distance of two (resulting in a $5\times5\times5$ cell region, or roughly 10% of each dimension). Part mixing is performed by replacing the corresponding coefficients in the base with those from the donor within the masked region, followed by the model's denoising and harmonization steps as described in the original work [13].

Proposed method: For each pair, we sample a camera position uniformly on the unit sphere, render a depth map of the donor, and randomly select a visible surface point as the center of the edit. We compute the nearest-neighbor distance from this donor point to the base surface, and use this value as the edit radius. All donor surface points within this radius are assigned a nonzero edit strength, which smoothly decays from 1 at the center to 0 at the boundary. For each donor surface point, the edit strength is determined by its normalized distance r to the edit center, using the template function $w(r) = \text{clamp}((1-r^4)\cdot\exp(-r^2), 0, 1)$. These spatially varying donor weights are input to our proposed model, generating a new edited shape where local donor features are transferred onto the base through local shape interpolation as described in Section 4.

Mesh Extraction and Metric Computation: For surface extraction, we scale the occupancy field outputs of SPAGHETTI [11] and our proposed method and apply a sigmoid activation, yielding a field normalized to the range $[-1, 1]$. Although this field is not a true SDF, it consistently indicates the zero level set and supports accurate surface extraction with marching cubes. Since any approximation error introduced by this normalization is present in both base and edited shapes, identical occupancy fields yield identical extracted surfaces in corresponding region. For Neural Wavelet, we use the generated TSDF grid directly for mesh extraction. Meshes for SPAGHETTI and Neural Wavelet are extracted using marching cubes on a 256^3 grid, following the original implementations. For our method, we have uses a 128^3 grid to reduce computation time since our method is significantly slower than the baseline methods. Meshes extracted at 128^3 and 256^3 resolutions appeared visually nearly identical, with 128^3 capturing all relevant details. This is consistent with the relatively low geometric detail of most ShapeNet [5] chair models, as well as the limitations of the generative models in representing fine geometric detail.. Even at lower resolutions, marching cubes extracts the surface based on the sampled occupancy values, so any change in these values, regardless of grid size, will displace the resulting surface coordinates. For each shape pair, we compute the edit locality metric as described in Section 5, varying the displacement threshold from 1% to 10% of the unit sphere radius in 1% increments.

6.3.2 Qualitative Visualization of Edit Propagation. To qualitatively evaluate the spatial propagation and magnitude of surface deformations introduced by different editing methods, we generate color-coded displacement maps for selected shape pairs. Surface vertices are first extracted by applying marching cubes to meshes reconstructed from 256^3-sampled TSDF's or occupancy fields as discussed in Section 6.3.1. To ensure dense sampling of the rendered surfaces, additional base surface points are sampled from the virtual camera rendering the displacements.

For each surface point on the edited shape sampled from the same camera, we compute its nearest-neighbor distance to the surface of the corresponding base, approximating it's surface point displacement from the unedited base.

To make the visualization more informative and easier to interpret, displacements below 1% of the unit sphere radius are masked (colored black), reflecting changes that are likely negligible for most applications and below our smallest threshold for measuring our locality metric. Displacements between 1% and 25% are mapped to a perceptually uniform color gradient, while displacements above 25% are limited at this value, resulting in clearer distinction between lower displacement values.

Each displacement measurement is performed on edits described in section 6.3.1 on one of the 100 shape pairs.

7 Limitations and Future Work

Since our method strongly depends on the quality of found semantic correspondences, inaccuracies in these correspondences can lead to artifacts or undesired deformations. Since we approximate features to align between base and donor shapes and the tangential component of the boundary sensitivity is neglected using boundary sensitivity [3], artifacts can form for large differences in structure. We attempted to implement the template learning methods proposed by Genova et al. [9] and Kim et al. [18] to improve semantic correspondence, but were unable to complete them due to time constraints.

Since our method builds on Spaghetti's [11] representation, mixing and interpolating disentangled normalized part geometry is non-trivial. This can result in significant displacements of selected local regions, including unintended translation, rotation, or scaling. Investigating mechanisms enabling separate interpolation of disentangled structure and geometry, could address these issues.

Although our proposed locality metric is simple, it does not account for the shape of the edited region. As a result, a long thin deformation region and a large uniform deformation region give the same value, even though the thin region is arguably more localized. Furthermore, since our experiments are conducted solely on the ShapeNet [5] chair category, the results may not generalize well to other categories with different shape distributions.

Our experiments are also fully automated and do not involve real-life human edits. While this enables consistent comparison of locality limits, it does not capture how users would perform and perceive local edits in practice. Finally, our work lacks both qualitative and quantitative metrics evaluating the visual quality and plausibility of the generated shapes, since no experiments were conducted to assess these properties.

8 Conclusion

In this work, we introduced a novel approach for local 3D shape editing that integrates semantic awareness from learning-based methods with theoretically unrestricted spatial locality. Our method represents shapes using a scalable, non-uniform, and arbitrarily dense set of blending weights combined with learned based shape representation SPAGHETTI [11]. This representation enables the transfer and interpolation of semantically meaningful surface features with high spatial precision, allowing for fine-grained and controllable edits.

We introduced a new metric to quantify edit locality and used it to empirically demonstrate that existing methods such as SPAGHETTI [11] and Neural Wavelet (2024) [13] exhibit significant edit propagation and coarse part region representation. Our experiments showed that our method enables more local edits compared to previous approaches. While challenges remain in semantic correspondence accuracy, dealing with varying structures and shape plausibility, our results provide a foundation for future exploration in semantically guided, fine-grained 3D shape manipulation.

References

[1] Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. 2021. imGHUM: Implicit Generative Models of 3D Human Shape and Articulated Pose. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 5441–5450. https://doi.org/10.1109/ICCV48922.2021.00541

[2] Daniele Baieri, Filippo Maggioli, Zorah Lähner, Simone Melzi, and Emanuele Rodolà. 2024. Implicit-ARAP: Efficient Handle-Guided Deformation of High-Resolution Meshes and Neural Fields via Local Patch Meshing. https://doi.org/10.48550/arXiv.2405.12895

[3] Arturs Berzins, Moritz Ibing, and Leif Kobbelt. 2023. Neural Implicit Shape Editing using Boundary Sensitivity. https://doi.org/10.48550/arXiv.2304.12951 arXiv:2304.12951 [cs]

[4] Mikolaj Binkowski, Danica J. Sutherland, Michal Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *International Conference on Learning Representations (ICLR)*, Vol. abs/1801.01401. https://openreview.net/forum?id=r1lUOzWCW

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. ShapeNet: An Information-Rich 3D Model Repository. https://doi.org/10.48550/arXiv.1512.03012

[6] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. 2003. On Visual Similarity Based 3D Model Retrieval. 22, 3 (2003), 223–232. https://doi.org/10.1111/1467-8659.00669

[7] Zhiqin Chen and Hao Zhang. 2019. Learning Implicit Fields for Generative Shape Modeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5932–5941. https://doi.org/10.1109/CVPR.2019.00609

[8] Yu Deng, Jiaolong Yang, and Xin Tong. 2021. Deformed Implicit Field: Modeling 3D Shapes with Learned Dense Correspondence. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 10281–10291. https://doi.org/10.1109/CVPR46437.2021.01015

[9] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. 2020. Local Deep Implicit Functions for 3D Shape. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4856–4865. https://doi.org/10.1109/CVPR42600.2020.00491

[10] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge Belongie. 2020. DualSDF: Semantic Shape Manipulation Using a Two-Level Representation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7628–7638. https://doi.org/10.1109/CVPR42600.2020.00765

[11] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. 2022. SPAGHETTI: editing implicit shapes through part aware generation. *ACM Transactions on Graphics (TOG)* 41, 4 (July 2022). https://doi.org/10.1145/3528223.3530084

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems* (2017), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html

[13] Jingyu Hu, Ka-Hei Hui, Zhengzhe Liu, Ruihui Li, and Chi-Wing Fu. 2024. Neural Wavelet-domain Diffusion for 3D Shape Generation, Inversion, and Manipulation. *ACM Transactions on Graphics* 43, 2 (January 2024). https://doi.org/10.1145/3635304

[14] Jingyu Hu, Ka-Hei Hui, Zhengzhe Liu, Hao Zhang, and Chi-Wing Fu. 2023. CLIPXPlore: Coupled CLIP and Shape Spaces for 3D Shape Exploration. In *SIGGRAPH Asia 2023 Conference Papers*. ACM, 1–12. https://doi.org/10.1145/3610548.3618144

[15] Jingyu Hu, Ka-Hei Hui, Zhengzhe Liu, Hao Zhang, and Chi-Wing Fu. 2024. CNSEdit: 3D Shape Editing via Coupled Neural Shape Optimization. *ACM Transactions on Graphics* 43, 4 (July 2024), 1 – 12. https://doi.org/10.1145/3641519.3657412

[16] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. 2022. Neural Wavelet-domain Diffusion for 3D Shape Generation. In *SIGGRAPH Asia 2022 Conference Papers*. ACM, 1–9. https://doi.org/10.1145/3550469.3555394

[17] Moritz Ibing, Isaak Lim, and Leif Kobbelt. 2021. 3D Shape Generation with Grid-based Implicit Functions. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 13554–13563. https://doi.org/10.1109/CVPR46437.2021.01335

[18] Sihyeon Kim, Minseok Joo, Jaewon Lee, Juyeon Ko, Juhan Cha, and Hyunwoo J Kim. 2023. Semantic-aware implicit template learning via part deformation consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 593–603.

[19] Xiaoyu Li, Qi Zhang, Di Kang, Weihao Cheng, Yiming Gao, Jingbo Zhang, Zhihao Liang, Jing Liao, Yan-Pei Cao, and Ying Shan. 2024. Advances in 3D Generation: A Survey. arXiv:2401.17807 [cs] http://arxiv.org/abs/2401.17807

[20] Connor Lin, Niloy Mitra, Gordon Wetzstein, Leonidas J Guibas, and Paul Guerrero. 2022. NeuForm: Adaptive Overfitting for Neural Shape Editing. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.). Curran Associates, Inc., 15217–15229.

[21] David Lopez-Paz and Maxime Oquab. 2017. Revisiting Classifier Two-Sample Tests. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJkXfE5xx

[22] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4455–4465. https://doi.org/10.1109/CVPR.2019.00459

[23] Marko Mihajlovic, Yan Zhang, Michael J. Black, and Siyu Tang. 2021. LEAP: Learning Articulated Occupancy of People. In *2021 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR). IEEE, 10456–10466. https://doi.org/10.1109/CVPR46437.2021.01032

[24] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. 2021. A-SDF: Learning Disentangled Signed Distance Functions for Articulated Shape Representation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 12981–12991. https://doi.org/10.1109/ICCV48922.2021.01276

[25] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 165–174. https://doi.org/10.1109/CVPR.2019.00025

[26] Zifan Shi, Sida Peng, Yinghao Xu, Andreas Geiger, Yiyi Liao, and Yujun Shen. 2022. Deep Generative Models on 3D Representations: A Survey. (2022). https://doi.org/10.48550/ARXIV.2210.15663

[27] J. Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 2022. 3D Neural Field Generation using Triplane Diffusion. arXiv:2211.16677 [cs] http://arxiv.org/abs/2211.16677

[28] Petros Tzathas, Petros Maragos, and Anastasios Roussos. 2023. 3D Neural Sculpting (3DNS): Editing Neural Signed Distance Functions. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 4510–4519. https://doi.org/10.1109/WACV56688.2023.00450

[29] Yun-Peng Xiao, Yu-Kun Lai, Fang-Lue Zhang, Chunpeng Li, and Lin Gao. 2020. A survey on deep geometry learning: From a representation perspective. *Computational Visual Media* 6, 2 (2020), 113–133. https://doi.org/10.1007/s41095-020-0174-8

[30] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. 2022. NeuMesh: Learning Disentangled Neural Mesh-Based Implicit Field for Geometry and Texture Editing. In *Computer Vision – ECCV 2022*. Vol. 13676. Springer Nature Switzerland, 597–614. https://doi.org/10.1007/978-3-031-19787-1_34

[31] Wang Yifan, Noam Aigerman, Vladimir G. Kim, Siddhartha Chaudhuri, and Olga Sorkine-Hornung. 2020. Neural Cages for Detail-Preserving 3D Deformations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 72–80. https://doi.org/10.1109/CVPR42600.2020.00015

[32] Yu-Jie Yuan, Yu-Kun Lai, Tong Wu, Lin Gao, and Ligang Liu. 2021. A Revisit of Shape Editing Techniques: From the Geometric to the Neural Viewpoint. *Journal of Computer Science and Technology* 36, 3 (June 2021). https://doi.org/10.1007/s11390-021-1414-9

[33] Mingwu Zheng, Hongyu Yang, Di Huang, and Liming Chen. 2022. Imface: A nonlinear 3d morphable face model with implicit neural representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 20343–20352.

[34] Xinyang Zheng, Yu Liu, Peng Wang, and Xin Tong. 2022. SDF-StyleGAN: Implicit SDF-Based StyleGAN for 3D Shape Generation. *Computer Graphics Forum* 41, 5 (2022), 52–63. https://doi.org/10.1111/cgf.14602