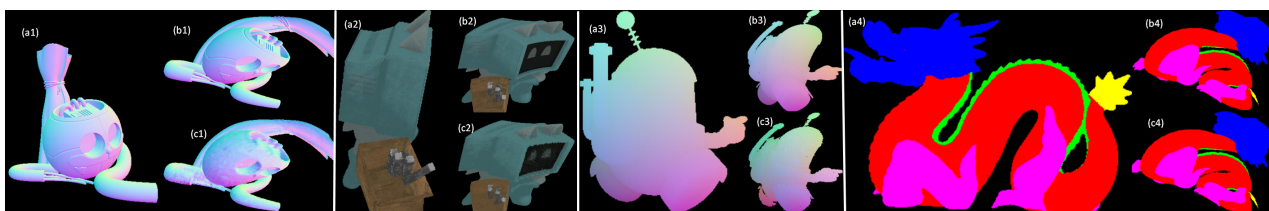


# Voxel Deformation-Aware Neural Intersection Function

Chih-Chen Kao , Grzegorz Makowski , Shin Fujieda , Takahiro Harada 

Advanced Micro Devices, Inc. (AMD)



**Figure 1:** Voxel Deformation-Aware Neural Intersection Function predicting geometric properties under parameterized deformation. From left to right: normal, albedo, distance, and material. For each example: (a) original shape, (b) deformed reference, (c) predicted result.

## Abstract

We extend the Locally-Subdivided Neural Intersection Function (LSNIF) to support parameterized deformable and animated geometry. Our approach introduces a rest-space and deformed-space formulation inspired by meshless rendering, allowing ray samples to be mapped back to a canonical space where a single neural network represents geometry consistently across poses without retraining. To maintain accuracy under deformation-aware training, we incorporate scale-invariant distance regression, uncertainty-weighted multi-task learning, and a hybrid positional-grid encoding. The resulting method preserves the compactness and efficiency of LSNIF while enabling robust neural intersection prediction for dynamic geometry.

## CCS Concepts

• Computing methodologies → Ray tracing; Neural networks;

## 1. Introduction

Neural representations for geometry and intersection queries have recently demonstrated impressive efficiency and accuracy, offering a promising alternative to traditional acceleration structures [MESK22]. Recent work such as N-BVH [WRM\*24] further explores hybrid approaches that combine neural ray queries with classical bounding volume hierarchies, highlighting a spectrum between learned and explicit acceleration structures. In particular, the Locally-Subdivided Neural Intersection Function (LSNIF) showed that object-space intersection queries can be represented by a compact neural network trained on voxelized geometry [FKH25]. By encoding spatial information using a sparse hash grid, LSNIF predicts intersection distance, surface normal, albedo, material, and occlusion for arbitrary query rays. This design enables accurate and memory-efficient intersection prediction while eliminating the need for a bottom-level Bounding Volume Hierarchy (BVH).

Despite these advantages, existing neural intersection methods, including LSNIF, are limited to static geometry. In practice, many

scenes contain deformable or articulated objects whose shape or pose varies over time. Handling deformation efficiently remains challenging, as neural networks trained on static geometry do not generalize to unseen poses without retraining. Related work in neural rendering has addressed dynamic scenes using deformation-aware implicit representations, such as D-NeRF [PCPMN21] and Deformable NeRF [PSB\*21], which learn a mapping between canonical and posed spaces. Extensions such as HyperNeRF [PSH\*21] further increase representational capacity for complex topological changes. However, these approaches focus on volumetric rendering rather than explicit ray-surface intersection queries, and are not directly designed for efficient object-space querying.

Classical graphics pipelines address deformation through skinning, meshless deformation, or volumetric warping by defining a mapping between a canonical rest space and a deformed space. However, incorporating such deformation models into neural implicit representations remains non-trivial, since geometry is encoded implicitly within network parameters.

© 2026 The Author(s).

Proceedings published by Eurographics - The European Association for Computer Graphics. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.2312/egs.20261004

To address this limitation, we extend the LSNIF framework to support parameterized deformation of voxelized geometry by adopting a rest-space and deformed-space formulation inspired by meshless rendering methods [LT25]. The neural network is trained exclusively in rest space, while ray tracing and data generation are performed in deformed space. Ray samples obtained in deformed space are mapped back to rest space before being encoded and processed by the network. This allows a single neural model to represent geometry consistently across multiple poses, enabling efficient and accurate intersection prediction for animated or deformable objects without per-pose retraining.

To maintain robustness when training across a wide range of deformations, we introduce deformation-aware training strategies that stabilize distance prediction, balance multiple geometric and appearance objectives, and improve spatial expressiveness under pose variation. In combination, these contributions yield a deformation-aware neural intersection function that preserves the compactness and efficiency of LSNIF while enabling robust and accurate intersection prediction for dynamic geometry.

## 2. Methodology

To support deformable or animated geometry within the LSNIF framework, a straightforward approach would be to treat each deformed configuration as an independent static object and train a separate neural intersection function per pose. While simple, this strategy is impractical in practice: it requires storing and training multiple models, scales poorly with the number of poses, and fails to exploit the strong geometric coherence shared across deformations. Moreover, even training a single model on multiple deformed shapes directly entangles pose variation with geometry, which significantly degrades accuracy.

To overcome these limitations, we adopt a two-space formulation: We define a *rest space*, which serves as the canonical coordinate system used to train the network, and a *deformed space*, which represents the geometry after a user-specified deformation is applied. The neural network always operates in rest space, while ray casting during dataset generation is performed in deformed space.

We voxelize the geometry in the rest space and apply parameterized deformation to obtain deformed voxels. For each deformation pose, we build a simple BVH over the deformed voxels and trace rays directly in this domain, where rays remain straight lines. Each ray-voxel intersection produces a hit point expressed in deformed-space coordinates. To make this information compatible with the rest-space LSNIF representation, every deformed-space hit point is transformed back into rest space using the inverse deformation.

Under an exact inverse mapping, a straight ray in deformed space corresponds to a curved path in rest space, and computing the intersection of such curved rays with rest-space geometry would be computationally expensive. In practice, we avoid this cost by applying a local linear approximation. Since LSNIF operates at the voxel level, we assume that within a single voxel, the deformation field is approximately linear. We therefore approximate the curved ray segment inside each voxel by a straight ray in rest space, which allows us to reuse standard ray-voxel and ray-geometry intersection routines. This approximation is efficient and empirically sufficient due

to the small spatial extent of individual voxels. Using this approximation, we intersect the locally linearized rays with the rest-space geometry to obtain ground-truth attributes for training, including hit distance, surface normal, albedo, material, and occlusion. By sampling a wide range of deformation parameters and repeating this process, we generate a dataset that associates deformed-space ray queries with rest-space geometric supervision.

During training, the transformed hit points in rest space are first encoded using the proposed input encoding described in Eq. 1. The retrieved grid features are concatenated and passed to a multi-layer perceptron (MLP), which predicts the same set of attributes as the original LSNIF. We employ task-specific loss terms and adopt an uncertainty-weighted multi-task formulation that dynamically adjusts the contribution of each loss, which will be discussed in the following subsections. This two-space deformation strategy enables a single neural model to remain valid across a wide range of poses. By mapping deformed-space ray queries back to a canonical rest-space representation and leveraging local linearization, the deformation-aware LSNIF retains the compactness and efficiency of the original method while providing stable and accurate intersection prediction for deformable and animated objects without per-pose retraining. The process is illustrated in Figure 2.

### 2.1. Voxel Traversal under Deformation

Voxel traversal is used to generate ray-voxel intersection samples for training. Unlike uniform grids, deformation induces spatially varying voxel sizes, which makes standard DDA traversal with a fixed step size invalid. Instead, traversal is performed by explicitly determining the voxel face through which a ray exits, which allows correct selection of neighboring voxels even under non-uniform or deformed voxel layouts.

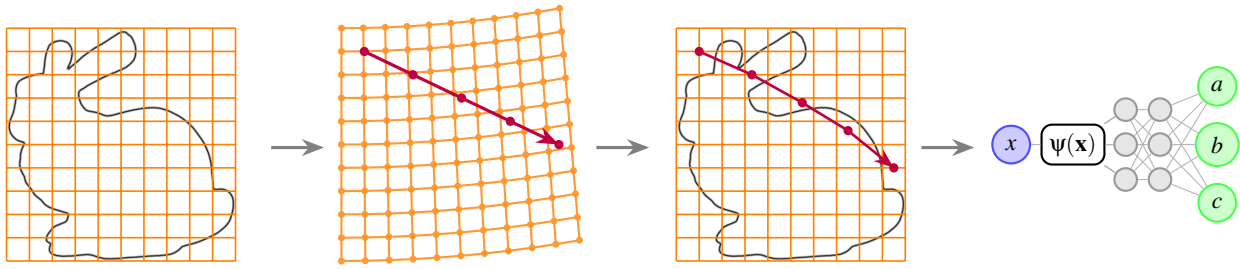
Another challenge is identifying the initial voxel containing the ray origin. Under deformation, voxel indices cannot be computed analytically from the ray origin due to non-uniform voxel extents. Rather than performing an expensive search, we initialize traversal by intersecting the ray with the occupied deformed voxels using a simple BVH. The first intersection point defines the starting voxel for traversal. Although this introduces an additional BVH query, its logarithmic cost is negligible compared to brute-force voxel lookup.

### 2.2. Segmented Ray Distance Accumulation

When a straight ray in deformed space is mapped back to rest space, it becomes piecewise linear across voxels. To compute intersection distances consistently, we accumulate the lengths of these ray segments.

Let  $(\mathbf{x}_i^0, \mathbf{x}_i^1)$  denote the entry and exit points of ray segment  $i$  within occupied voxel  $i$  in deformed space, with segment length  $\ell_i = \|\mathbf{x}_i^1 - \mathbf{x}_i^0\|$  and inter-voxel offset  $\delta_i = \|\mathbf{x}_i^0 - \mathbf{x}_{i-1}^1\|$ . The local hit distance  $t_i \in [0, \ell_i]$  is measured from the segment entry point. The total intersection distance is then:

$$D = \sum_{j=0}^{k-1} (\delta_j + \ell_j) + \delta_k + t_k,$$



**Figure 2:** Deformation-aware training, from left to right: (a) the rest-space voxel grid (b) the warped deformed-space grid with a straight ray (c) the piecewise linear approximation in rest space (d) index  $x$  fed to the Eq. 1 encoder and MLP to predict geometric properties  $(a, b, c)$ .

where  $k$  indexes the first voxel containing a hit.

During evaluation, we reconstruct hit positions along the same segmented ray in rest space. This formulation ensures that distance predictions are consistent with spatial localization and simplifies supervision for deformation-aware training.

### 2.3. Combining Positional Encoding with the Grid Encoding

LSNIF represents spatial locations using a multi-resolution hash grid encoding, which provides a compact and efficient alternative to dense voxel grids. While effective for capturing localized geometric detail, we observe that under deformation, relying solely on grid-based features can limit the network’s ability to model smooth global variations and high-frequency spatial signals consistently across poses. To address this limitation, we augment the original LSNIF representation with an explicit positional encoding and combine it with the grid encoding in a joint framework.

Given a 3D query point  $\mathbf{x} \in \mathbb{R}^3$ , we apply a sinusoidal positional encoding  $\gamma(\mathbf{x})$  defined in Eq. 1. The encoding maps each spatial coordinate to a set of sinusoidal functions with exponentially increasing frequencies, where  $F$  denotes the number of frequency bands. This formulation injects multi-scale spatial information and enables the network to represent both low-frequency global structure and high-frequency surface detail.

In parallel, we employ the multi-level hash grid encoding. Given a normalized query point  $\mathbf{x} \in [0, 1]^3$ , features are queried from a hierarchy of hash grids with exponentially increasing resolutions. At each level  $l$ , features are interpolated from neighboring grid vertices and aggregated across levels via concatenation, as formalized in Eq. 1, where  $\phi(\cdot)$  denotes the grid encoder.

We combine positional encoding and hash grid encoding through the composite mapping  $\psi(\mathbf{x})$  defined in Eq. 1 [PMH26]. Specifically, each query point is first expanded via positional encoding, and the resulting frequency-modulated samples are independently processed by the multi-level hash grid encoder. The encoded features are then flattened and passed to the neural intersection network. This joint encoding strategy preserves the locality and memory efficiency of hash grids while retaining the expressive power of explicit sinusoidal features. Empirically, we find that this combination improves robustness under deformation and leads to more stable and accurate prediction of intersection-related attributes across varying poses.

$$\begin{aligned} \gamma(\mathbf{x}) &= \left[ \mathbf{x}, \left\{ \sin(2^k \mathbf{x}), \cos(2^k \mathbf{x}) \right\}_{k=0}^{F-1} \right], \\ \phi(\mathbf{x}) &= \bigoplus_{l=1}^L \phi_l(\mathbf{x}), \quad \psi(\mathbf{x}) = \phi(\gamma(\mathbf{x})) \end{aligned} \quad (1)$$

### 2.4. Uncertainty-Weighted Multi-Task Loss

We adopt an uncertainty-weighted multi-task objective [CGK18], where each task has a learnable log-variance  $s_i = \log \sigma_i^2$  that dynamically balances its contribution. Regression tasks use  $\mathcal{L}_i^{\text{reg}} = 0.5 \exp(-s_i) \mathcal{L}_i + 0.5 s_i$ , classification tasks use  $\mathcal{L}_i^{\text{cls}} = \exp(-s_i) \mathcal{L}_i + 0.5 s_i$ , and the total loss sums over all tasks.

For distance prediction, deformations produce multi-segment rays, so accumulated distance is unbounded. We replace the sigmoid output with ReLU and regress log-transformed distances using MSE in log space. Surface normals use a cosine similarity loss, albedo is predicted via sigmoid RGB values with  $\ell_1$  loss; both as regression tasks. Material is multi-class, occlusion binary; both follow the classification weighting  $\mathcal{L}^{\text{cls}}$ .

## 3. Experimental Results and Discussion

### 3.1. Accuracy Comparison without Deformation

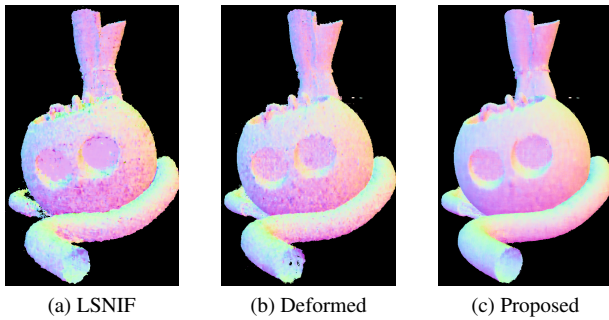
To isolate the effect of deformation-aware training, we evaluate both our method and the original LSNIF on static, single-pose geometry. No deformation is applied, so differences arise solely from training strategy. As expected, the original LSNIF slightly outperforms in this setting, since training on a single pose allows overfitting to pose-specific geometry and appearance, yielding lower distance and normal errors (Table 1). Our deformation-aware model, trained on a pose distribution, is regularized and cannot fully specialize to a single configuration. Nevertheless, the proposed modifications reduce this gap, and in some metrics our model even surpasses LSNIF. Figure 3 shows a visual comparison.

### 3.2. Comparison of Deformation Results

We compare a deformation-aware LSNIF baseline without our modifications to the full model incorporating all proposed components. Both are trained on the same deformed poses and evaluated on identical ray queries with voxel resolution  $32^3$ , a two-

**Table 1:** Prediction error on a single static pose. *LSNIF* denotes the original method trained on a single pose. *Deformed* is a deformation-aware baseline without the proposed training and encoding modifications. *Proposed* applies all proposed methods.

Method	Distance (mean-absolute-error)	Normal ( $^{\circ}$ )
LSNIF	0.105	19.1
Deformed	0.135	24.7
Proposed	<b>0.039</b>	<b>11.8</b>



**Figure 3:** Single-pose comparison: (a) original LSNIF, (b) deformation-aware without our modifications, (c) deformation-aware with all proposed methods.

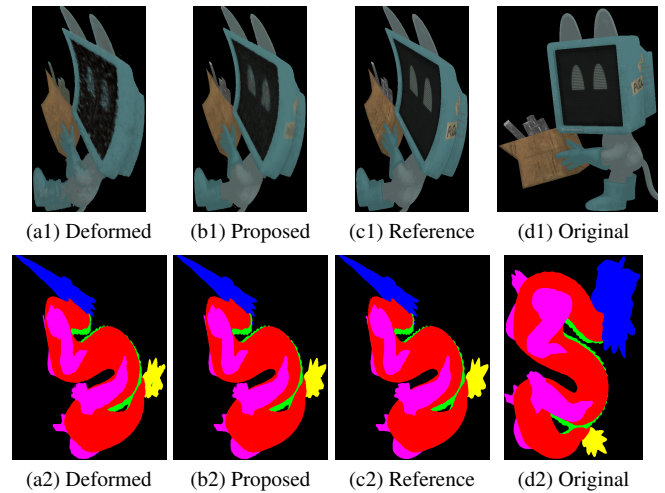
level hash grid (base 64, feature dim 3), and  $F = 3$  positional encoding bands. By employing a simple BVH over deformed voxels and trace straight rays directly in this domain, where rays remain straight; this adds only 2–3% computational overhead. As shown in Figure 4, the baseline suffers degraded intersection accuracy and appearance consistency under deformation, with blurred surfaces and unstable attributes. The full model produces sharper, consistent geometry, demonstrating the importance of our deformation-aware extensions for robust neural intersection prediction.

#### 4. Conclusion, Limitation, and Future Work

We presented a deformation-aware extension of LSNIF that enables efficient neural intersection prediction for parameterized deformable geometry. By mapping ray queries from deformed space to a canonical rest space, a single neural model generalizes across multiple poses without per-pose retraining. Curved rays induced by deformation are approximated using voxel-wise linear segments, preserving accuracy and efficiency.

To improve stability under pose variability, we introduced log-scaled distance regression, uncertainty-weighted multi-task learning, and a hybrid positional-grid encoding. Experiments show that neural intersection functions can be extended beyond static geometry while retaining LSNIF’s compactness.

While our approach handles articulated and simple morphing deformations, highly non-linear, topologically changing, or physics-driven deformations may require more expressive representations or adaptive voxelization. Future work includes supporting these general deformations, learning deformation fields jointly with intersection prediction, and integrating the method into full rendering pipelines.



**Figure 4:** Comparison of deformation-aware predictions across two scenes. Top row: albedo prediction on the cat object. Bottom row: multi-material prediction on the dragon object.

#### References

- [CGK18] CIPOLLA R., GAL Y., KENDALL A.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 7482–7491. doi:10.1109/CVPR.2018.00781. 3
- [FKH25] FUJIEDA S., KAO C.-C., HARADA T.: Lsnif: Locally-subdivided neural intersection function. *Proc. ACM Comput. Graph. Interact. Tech.* 8, 1 (May 2025). URL: <https://doi.org/10.1145/3728295>, doi:10.1145/3728295. 1
- [LT25] LUTON P., TRICARD T.: Real-time rendering of animated meshless representations. In *HPG 2025 - Conference on High Performance Graphics* (Copenhagen, Denmark, June 2025), pp. 1–9. URL: <https://hal.science/hal-05095359>. 2
- [MESK22] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* 41, 4 (July 2022). URL: <https://doi.org/10.1145/3528223.3530127>, doi:10.1145/3528223.3530127. 1
- [PCPMN21] PUMAROLA A., CORONA E., PONS-MOLL G., MORENO-NOGUER F.: D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 10318–10327. 1
- [PMH26] PEREZ G., MATAI J., HARADA T.: Peps: Positional encoding projected sampling. *Proc. ACM Comput. Graph. Interact. Tech.* To Appear (2026). 3
- [PSB\*21] PARK K., SINHA U., BARRON J. T., BOUAZIZ S., GOLDMAN D. B., SEITZ S. M., MARTIN-BRUALLA R.: Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 5865–5874. 1
- [PSH\*21] PARK K., SINHA U., HEDMAN P., BARRON J. T., BOUAZIZ S., GOLDMAN D. B., MARTIN-BRUALLA R., SEITZ S. M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228* (2021). 1
- [WRM\*24] WEIER P., RATH A., MICHEL É., GEORGIEV I., SLUSALLEK P., BOUBEKEUR T.: N-bvh: Neural ray queries with bounding volume hierarchies. In *ACM SIGGRAPH 2024 Conference Papers* (2024), pp. 1–11. 1