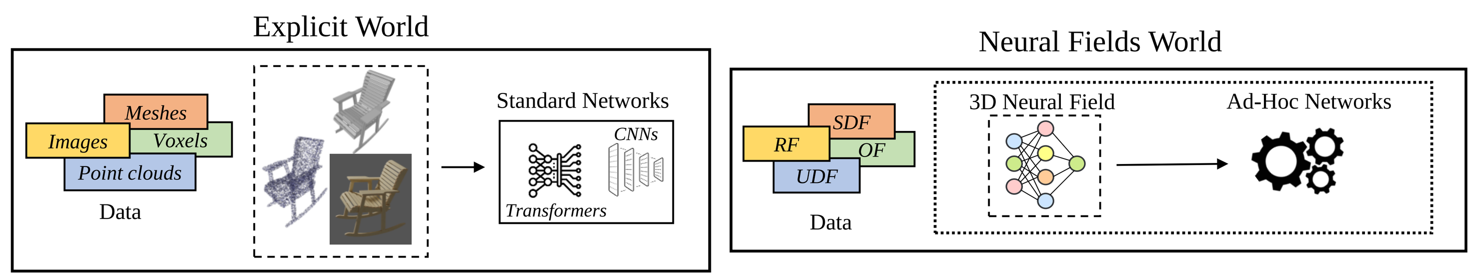
Neural Processing of Tri-Plane Hybrid Neural Fields

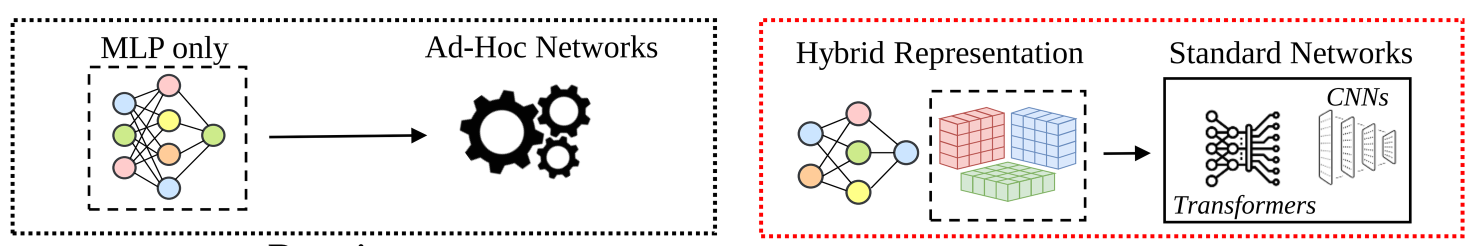
Adriano Cardace[1], Pierluigi Zama Ramirez[1], Francesco Ballerini[1]
Allan Zhou[2], Samuele Salti[1], Luigi Di Stefano[1]
[1]University of Bologna, Department of Computer Science and Engineering
[2]Stanford University

Motivation

Explicit World vs Neural Field World. 3D Neural Fields are functions defined at all spatial coordinates, parameterized by a neural network such as a Multi-Layer Perceptron (MLP), used to represent the 3D world around us. Various types of 3D neural fields have been explored, such as the signed/unsigned distance field (*SDF*/*UDF*), the occupancy field (*OF*), and the radiance field (*RF*). They have several advantages over discrete representations such as voxels, mesh, or point clouds: • they provide a continuous representation of the world; • they encode a 3D geometry at arbitrary resolution while using a finite number of parameters (the weights of the MLP).
3D neural fields may become one of the standard methods for storing and communicating 3D information; thus, developing strategies to solve tasks such as classification or segmentation by directly processing neural fields is relevant as well.



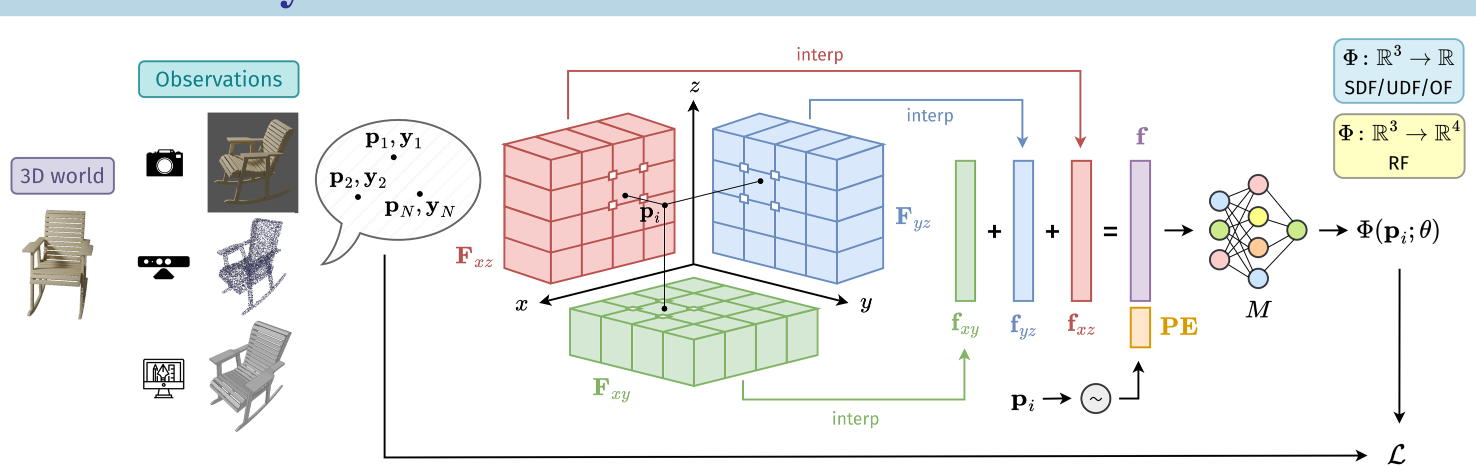
Hybrid neural fields combine continuous neural elements (i.e., MLPs) with discrete spatial structures (e.g., voxel grids) that encode local information for faster inference, better use of network capacity and suitability for editing tasks. This paper shows that the discrete features encode rich semantic and geometric information, which can be processed by applying well-established architectures. Moreover, we note how similar information is stored in tri-planes with different initializations of the same shape. Yet, the information is organized with different channel orders. For this reason, we propose to employ architectures invariant to the channel order, and we are able to achieve good performance in the challenging but more realistic scenario of randomly initialized neural fields. Indeed, we almost close the gap between methods that operate on explicit representations and those working directly on neural representations.



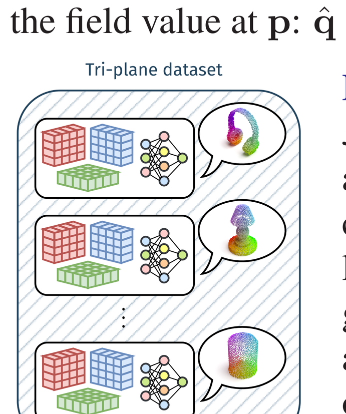
Previous

Ours

Tri-Plane Hybrid Neural Field



A special case of hybrid neural fields is parameterized by a discrete tri-plane feature map, T, and a small MLP network, M. T consists of three orthogonal 2D feature maps, $T = (\mathbf{F}_{xy}, \mathbf{F}_{xz}, \mathbf{F}_{yz})$, with $\mathbf{F}_{xy}, \mathbf{F}_{xz}, \mathbf{F}_{yz} \in \mathbb{R}^{C \times H \times W}$, where C is the number of channels and W, H are the spatial dimensions of the feature maps. The feature vector associated with a 3D point, $\mathbf{p}$, is computed by projecting the point onto the three orthogonal planes so to get the 2D coordinates, $\mathbf{p}_{xy}$, $\mathbf{p}_{xz}$, and $\mathbf{p}_{yz}$, relative to each plane. Then, the four feature vectors corresponding to the nearest neighbours in each plane are bi-linearly interpolated to calculate three feature vectors, $\mathbf{f}_{xy}$, $\mathbf{f}_{xz}$, and $\mathbf{f}_{yz}$, which are summed up element-wise to obtain $\mathbf{f} = \mathbf{f}_{xy} + \mathbf{f}_{xz} + \mathbf{f}_{yz}$, $\mathbf{f} \in \mathbb{R}^C$. Finally, we concatenate $\mathbf{f}$ with a positional encoding, $\mathbf{PE}$, of the 3D point $\mathbf{p}$ and feed it to the MLP, which in turn outputs the field value at $\mathbf{p}$: $\hat{q} = \Phi(\mathbf{p};\theta) = M([\mathbf{f}, \mathbf{PE}])$.

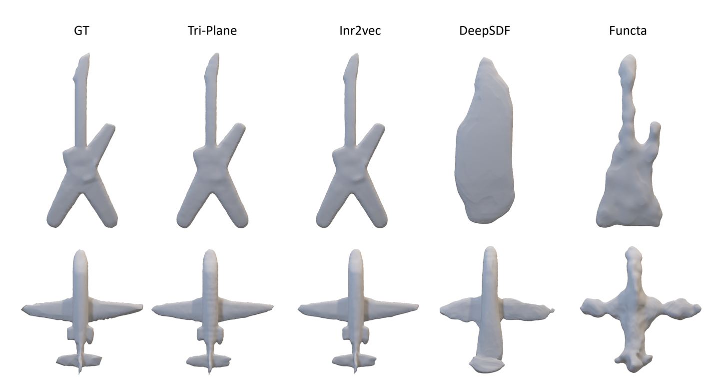


Learning tri-planes. To learn a field, we optimize a (T, M) pair *for each 3D object*, starting from randomly initialized parameters, θ, for both M and T. We sample N points $\mathbf{p}_i$ and feed them to T and M to compute the corresponding field quantities $\hat{q}_i = \Phi(\mathbf{p}_i;\theta)$. Then, we optimize θ with a loss, $\mathcal{L}$, capturing the discrepancy between the predicted fields $\hat{q}_i$ and the ground truth $\mathbf{y}_i$, applying an optional mapping between the output and the available supervision if needed. We repeat this process for each 3D shape of a dataset, thereby creating a dataset of tri-plane hybrid neural fields.

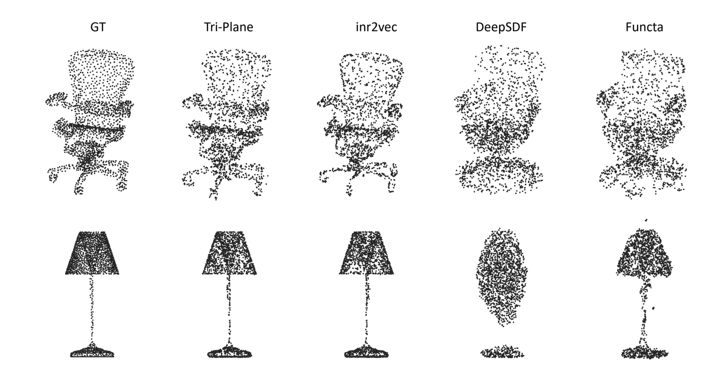
Tri-Plane Reconstruction Quality

Method	Type	# Params (K) ↓	Mesh from *SDF*: CD (mm) ↓	Mesh from *SDF*: F-score (%) ↑	Point Cloud from *UDF*: CD (mm) ↓	Point Cloud from *UDF*: F-score (%) ↑
inr2vec	Single	800	0.26	**69.7**	**0.21**	**65.5**
Tri-plane	Single	**64**	**0.18**	68.6	0.24	60.7
Tri-plane	Shared	**64**	1.57	42.9	3.45	33.3
DeepSDF	Shared	2400	6.6	25.1	5.6	5.7
Functa	Shared	7091	2.85	21.3	12.8	5.8

Mesh and point cloud reconstruction results on the Manifold40 test set. "Single" and "Shared" indicate neural fields trained on each shape independently or on the whole dataset.

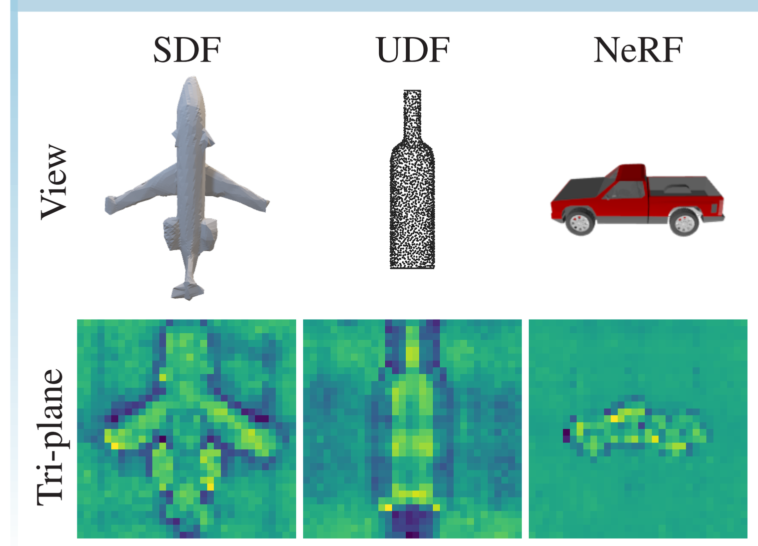


Reconstruction comparison for Manifold40 meshes obtained from *SDF*



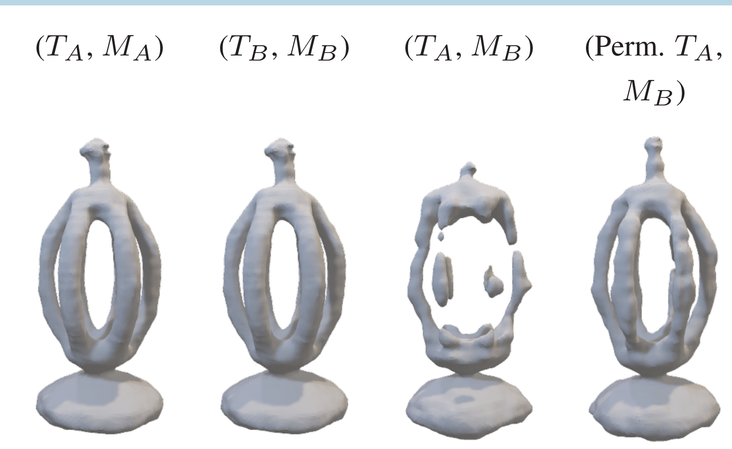
Reconstruction comparison for ModelNet40 point clouds obtained from *UDF*

Tri-Plane Content

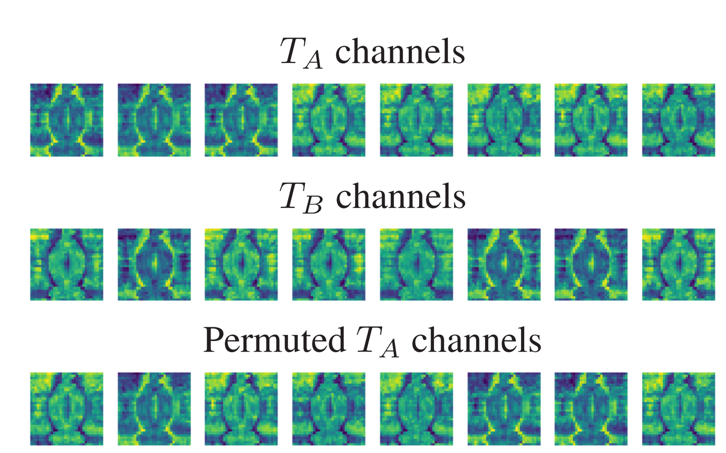


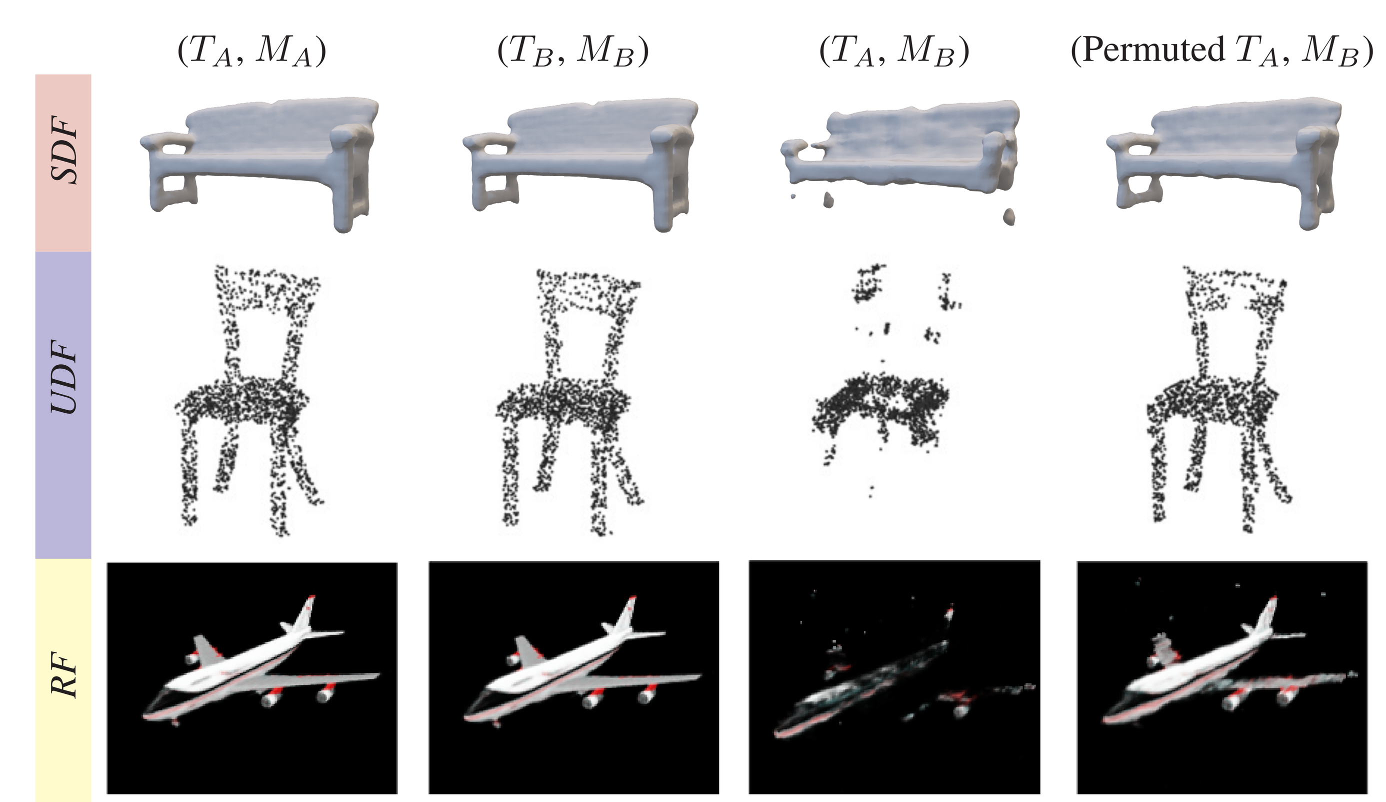
We render a view of the reconstructed 3D object alongside the corresponding tri-plane feature map for three different hybrid neural fields. These visualizations show clearly that the tri-plane spatial structure learns the object shape, i.e., it contains information about its geometry. For this reason, when tackling tasks such as classification and segmentation, we discard the MLPs and process only the tri-plane structure of the neural fields.

Tri-Plane Random Initialization

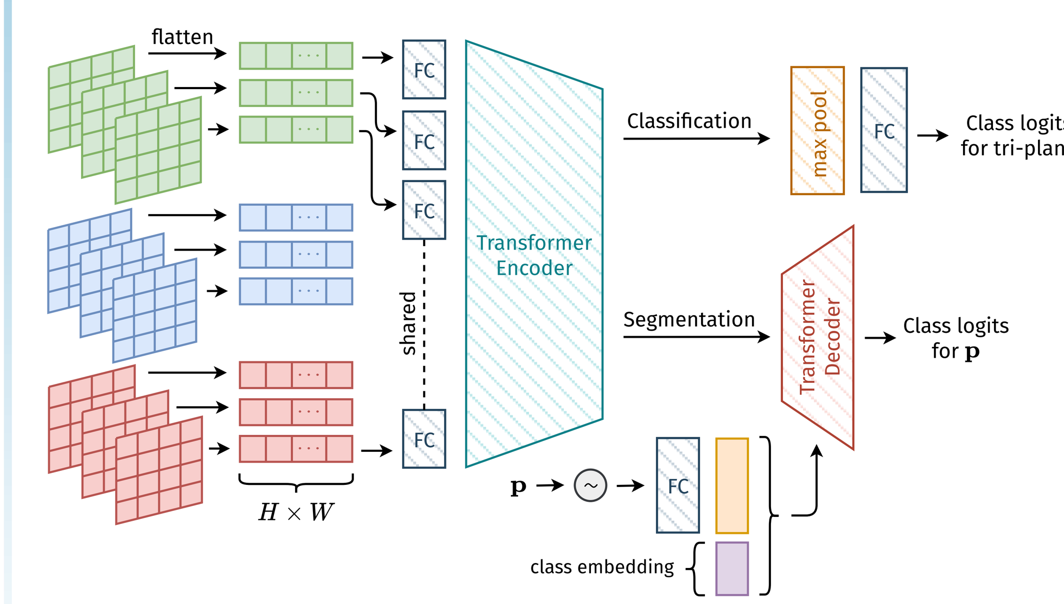


Reconstructions of different combinations of (tri-plane, MLP) pairs with different inits





Architecture for Tri-Plane Processing



We process tri-planes with a Transformer encoder without positional encoding, which is equivariant to token positions. As each token represents a channel of a plane, our architecture computes representations equivariant to the order of the channels. We unroll each channel of size $H \times W$, to obtain a token of dimension HW within a sequence of length $3C$ tokens. These tokens are then linearly projected and fed into the Transformer. The output of the encoder is once again a sequence of $3C$ tokens. The output sequence is subjected to a max pool operator for global tasks like classification to obtain a global embedding that characterizes the input shape. The way the tokens are defined, the absence of positional encoding, and the final max pool operator allow for achieving invariance to the channel order. We also utilize the decoder part of Transformers for dense tasks like part segmentation. Specifically, we treat the coordinates queries to segment as a sequence of input tokens to the decoder. Each point $\mathbf{p}$ with coordinates (x, y, z) undergoes positional encoding and is then projected to a higher-dimensional space using a linear layer. By leveraging the cross-attention mechanisms within the decoder, each input token representing a query point can globally attend to the most relevant parts of the tri-planes processed by the encoder to produce its logits.

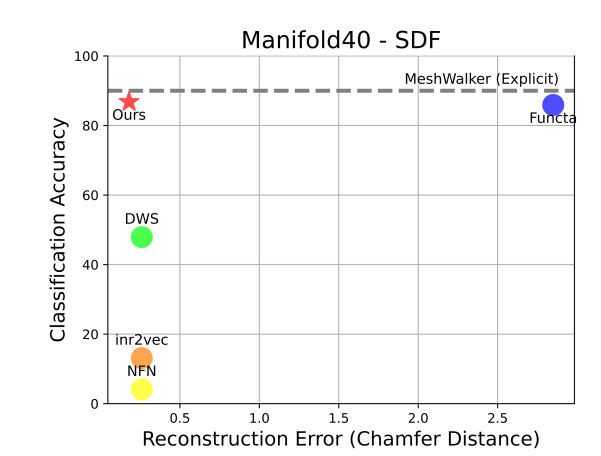
Neural Field Classification

Method	Type	Input	*UDF*: ModelNet40	*UDF*: ShapeNet10	*UDF*: ScanNet10	*SDF*: Manifold40	*OF*: ShapeNet10	*RF*: ShapeNetRender
DeepSDF	Shared	Latent vector	41.2	76.9	51.2	64.9	–	–
Functa	Shared	Modulation	**87.3**	83.4	56.4	85.9	36.3	–
inr2vec	Single	MLP	10.6	42.0	40.9	13.1	38.6	–
MLP	Single	MLP	3.7	28.8	36.7	4.2	29.6	22.0
NFN	Single	MLP	9.0	9.0	45.3	4.1	33.8	87.0
NFT	Single	MLP	6.9	6.9	45.3	4.1	33.8	85.3
DWSNet	Single	MLP	56.3	78.4	62.2	47.9	79.1	83.1
Ours	Single	Tri-plane	87.0	**94.1**	**69.1**	**86.8**	**91.8**	**92.6**

Test set accuracy for shape classification across neural fields. We compare several frameworks capable of processing neural fields.

Method	Input	ModelNet40	ShapeNet10	ScanNet10	Manifold40	ShapeNet10	ShapeNetRender
Ours	Tri-plane	87.0	94.1	69.1	86.8	91.8	92.6
PointNet	Point Cloud	**88.8**	**94.3**	**72.7**	–	–	–
MeshWalker	Mesh	–	–	–	**90.0**	–	–
Conv3DNet	Voxel	–	–	–	–	**92.1**	–
ResNet50	Images	–	–	–	–	–	**94.0**

Comparison with explicit representations. Top: Test set accuracy of our neural field processing method. **Bottom:** Standard networks trained and tested on explicit representations.



Train: *UDF*	Train: *SDF*	Train: *OF*	Test: *UDF*	Test: *SDF*	Test: *OF*
✓			84.7	78.4	15.6
	✓		67.3	86.8	11.9
		✓	49.3	46.9	77.7
✓	✓	✓	**87.4**	**87.8**	**80.3**

Universal tri-plane classifier. Test set accuracy on Manifold40.

Neural Field 3D Part Segmentation

Method	Input	instance mIoU	class mIoU	airplane	bag	cap	car	chair	earphone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skateboard	table
inr2vec	MLP	64.2	64.5	57.9	72.9	67.8	56.4	67.6	48.4	81.6	70.6	55.5	88.8	51.5	87.2	64.7	40.1	58.4	62.5
Ours	Tri-plane	**84.2**	**81.3**	**83.0**	**80.2**	**87.4**	**76.6**	**90.2**	**68.2**	**91.6**	**85.9**	**82.1**	**95.0**	**70.7**	**94.4**	**81.9**	**59.0**	**73.4**	**80.9**
PointNet	Point Cloud	83.1	78.96	81.3	76.9	79.6	71.4	89.4	67.0	91.2	80.5	80.0	95.1	66.3	91.3	80.6	57.8	73.6	81.5
PointNet++	Point Cloud	84.9	82.73	82.2	88.8	84.0	76.0	90.4	80.6	91.8	84.9	84.4	94.9	72.2	94.7	81.3	61.1	74.1	82.3
DGCNN	Point Cloud	83.6	80.86	80.7	84.3	82.8	74.8	89.0	81.2	90.1	86.4	84.0	95.4	59.3	92.8	77.8	62.5	71.6	81.1

Part segmentation results. Top: Implicit frameworks. **Bottom:** Methods on explicit representation. In **bold**, best results among frameworks processing neural fields.

Ablation Studies

Method	Input	*UDF*: ModelNet40	*UDF*: ShapeNet10	*UDF*: ScanNet10	*SDF*: Manifold40	*OF*: ShapeNet10
MLP	Tri-plane	41.6	84.2	55.8	40.2	79.1
CNN	Tri-plane	82.2	92.1	63.4	82.5	88.4
PointNet	Tri-plane	85.8	93.4	**69.3**	85.6	91.5
Spatial PointNet	Tri-plane	32.3	65.4	51.3	37.0	54.7
Transformer	Tri-plane	**87.0**	**94.1**	69.1	**86.8**	**91.8**

Ablation study of architectures for tri-plane neural field classification

Method	Input	*SDF*: Manifold40
MLP	MLP (Tri-plane)	4.3
NFN	MLP (Tri-plane)	4.1
NFT	MLP (Tri-plane)	4.1
inr2vec	MLP (Tri-plane)	7.4
Ours	Tri-plane	**86.8**

Classification of MLPs of tri-plane neural fields on the Manifold40 test set. Neural fields were randomly initialized.

Method	Type	*SDF* Classification: Accuracy (%) ↑	Mesh from *SDF* Reconstruction: CD (mm) ↓	Mesh from *SDF* Reconstruction: F-score (%) ↑
Tri-plane	Shared	84.7	1.57	42.9
Tri-plane (Ours)	Single	**86.8**	**0.18**	**68.6**

Shared vs individual MLP. Comparison of classification and reconstruction results of tri-planes sharing all the same MLP vs when each tri-plane has its own individual MLP. Results were computed on the Manifold40 test set.

Resolution	Channels	*SDF*: Accuracy (%) ↑	Mesh from *SDF*: CD (mm) ↓	Mesh from *SDF*: F-score (%) ↑
32 × 32	32	86.3	0.18	68.6
32 × 32	16	**86.8**	0.18	68.8
32 × 32	8	86.4	0.18	**69.2**
24 × 24	16	86.6	0.18	68.9
40 × 40	16	86.4	0.18	69.0

Ablation study of tri-plane resolution and number of channels. Second row is our choice. Results on the Manifold40 test set.