

Supplementary material for: Learnable Earth Parser: Discovering 3D Prototypes in Aerial Scans

Romain Loiseau^{1,2}
romain.loiseau@enpc.fr

Elliot Vincent^{1,3}
elliott.vincent@enpc.fr

Mathieu Aubry¹
mathieu.aubry@enpc.fr

Loic Landrieu^{1,2}
loic.landrieu@enpc.fr

¹ LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, France

² Univ Gustave Eiffel, IGN, ENSG, LASTIG, France

³ INRIA Paris, France

In this supplementary material, we provide details on the implementation of our Learnable Earth Parser (Sec. A-1), details about our proposed Earth Parser Dataset (Sec. A-2), and some additional quantitative (Sec. A-3) and qualitative (Sec. A-4) results. Our code and dataset are available at <https://romainloiseau.fr/learnable-earth-parser/>.

A-1. Detailed Configuration

We report here the exact architecture of the Learnable Earth Parser network and training details.

Learnable prototypes. Following Loiseau *et al.* [5], the point coordinates of our prototypes $\mathbf{P}^1, \dots, \mathbf{P}^K$ are learned directly as free parameters of the model through our reconstruction loss. Each prototype contains 256 points leading to learning $K \times 256 \times 3$ free parameters to represent all the learned prototypes. Eventually, our model’s 3D prototypes are defined by their points’ coordinates, which are free parameters learned by optimizing the reconstruction loss \mathcal{L}_{rec} and its regularization \mathcal{L}_{reg} . While the reconstruction task serves as a label-free supervisory signal, our main goal is not to achieve the best possible reconstruction but to learn simple and interpretable prototypes. A model using feature-space prototypes and arbitrary transformations may achieve a much lower reconstruction error, but its prototypes would have low semantic purity and interpretability.

Network architecture. Our model takes a point cloud \mathbf{X} and computes a voxelization in a grid of size $64 \times 64 \times 64$. As shown in Figure A-1, our model is composed of (i) a point encoder $\mathcal{E}_{\text{point}}$, (ii) a scene encoder $\mathcal{E}_{\text{scene}}$, (iii) S slot feature extractors \mathcal{D}_s and (iv) five shared slot parameters generators: $\mathcal{D}_{\text{proba}}$, $\mathcal{D}_{\text{scale}}$, $\mathcal{D}_{\text{rot-y}}$, $\mathcal{D}_{\text{rot-z}}$, $\mathcal{D}_{\text{translate}}$. We provide details on these networks below.

• **Point encoder.** Each input point of \mathbf{X} is associated with a 10-dimensional descriptor: (1-3) normalized position in the tile in $[-1, 1]^3$, (4-6) *rgb* color, (7) normalized LiDAR reflectance, and (8-10) its offset relative to the center of its assigned voxel. The point encoder $\mathcal{E}_{\text{point}}$ is a linear layer that maps these descriptors to a 16-dimensional point feature.

• **Scene encoder.** We compute voxel features by max-pooling the features of the points associated to each voxel. The scene encoder $\mathcal{E}_{\text{scene}}$ then maps these voxel features to a single scene feature, a vector of size 1024, by using a sequence of 6 3D sparse convolutions [2] with kernel size $[3, 3, 3]$ and 6 strided convolutions with kernel size $[2, 2, 2]$ and stride $[2, 2, 2]$.

• **Slot feature extractor.** Each slot s takes as input the scene feature produced by $\mathcal{E}_{\text{scene}}$ and maps it to a slot feature of size 128 with a dedicated linear layer \mathcal{D}_s .

• **Slot parameters generators.** Five 3-layers Multi Layer Perceptrons (MLPs) are shared by all slots to map their slot features to the associated parameters of the reconstruction model.

- $\mathcal{D}_{\text{proba}}$ outputs the slot activation and prototype choice probability α_s et β_s^k .
- $\mathcal{D}_{\text{scale}}$ outputs three scales in $[-1/2, 2]$, corresponding to scaling the prototypes in each canonical directions.
- $\mathcal{D}_{\text{rot-y}}$ outputs a rotation in $[-\pi/10, \pi/10]$ to be applied around the y axis.
- $\mathcal{D}_{\text{rot-z}}$ outputs a 2D point on the unit circle which is then mapped to a rotation in $[-\pi, \pi]$ to be applied around the z axis.
- $\mathcal{D}_{\text{translate}}$ outputs a 3D translation vector in \mathbb{R}^3 .

These parameters are used to determine the activation of the slot, choose a prototype, then apply a sequence of transformations in the following order: scaling, y -rotation, z -rotation, and translation.

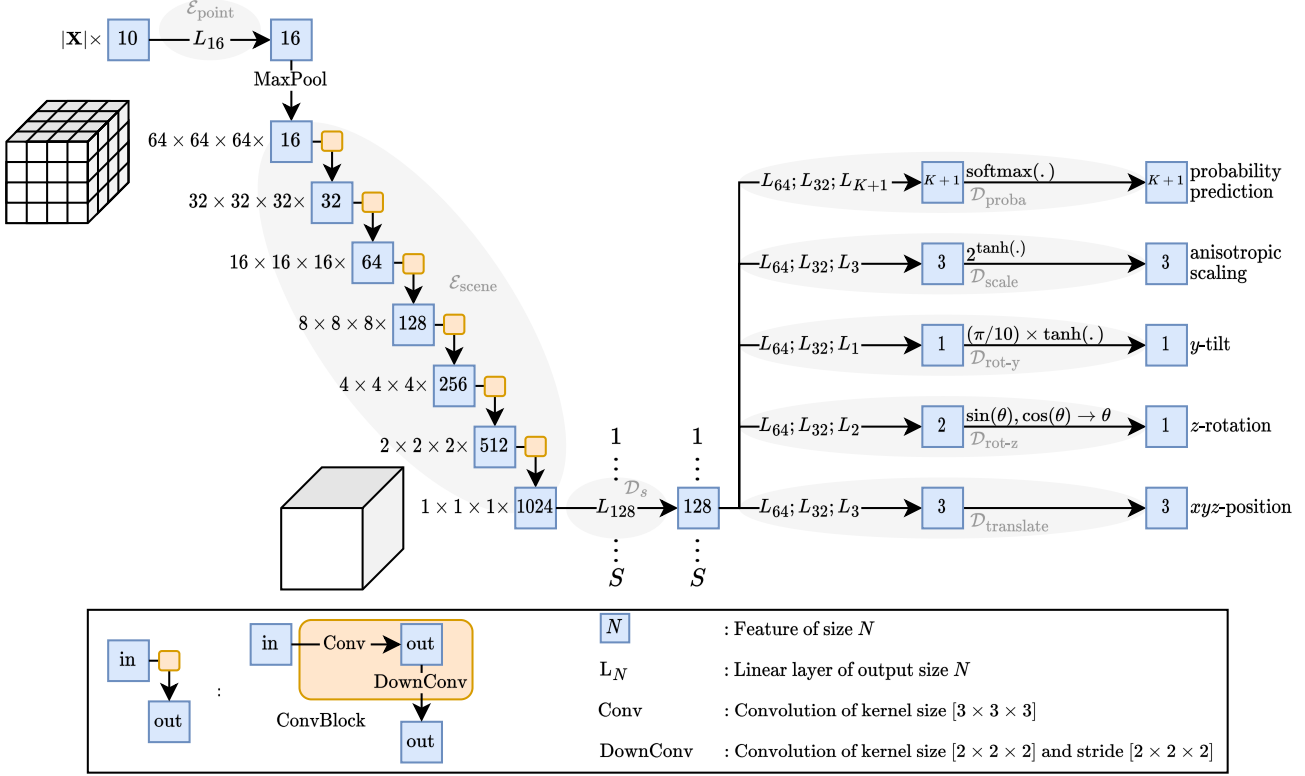


Figure A-1. **Learnable Earth Parser Detailed Architecture.** Details of the architecture showing all layers in $\mathcal{E}_{\text{point}}$, $\mathcal{E}_{\text{scene}}$, \mathcal{D}_s , $\mathcal{D}_{\text{proba}}$, $\mathcal{D}_{\text{scale}}$, $\mathcal{D}_{\text{rot-y}}$, $\mathcal{D}_{\text{rot-z}}$ and $\mathcal{D}_{\text{translate}}$. We use LayerNorm [1] and LeakyRelu after all hidden layers.

Reconstruction loss. Due to the arbitrary square shape of our samples \mathbf{X} , some objects can appear only partly in a patch. We don’t want the network to learn prototypes specifically to fit such object parts, as it is an artifact of our sampling procedure. Indeed, square patches are sampled randomly during training, and along a non-overlapping grid for inference. Instead, we propose to ignore the points of the reconstruction \mathbf{Y}_s^k that falls beyond the normalized $[-1, 1]$ extent of the patches. This allows the network to predict full objects without being penalized in terms of accuracy. To do so, we modify Equation 8 from the main paper as follows:

$$\mathcal{L}_{\text{acc}}(\mathcal{M}, \mathbf{X}) = \frac{1}{S} \sum_{s=1}^S \sum_{k=1}^K \beta_s^k d(\tilde{\mathbf{Y}}_s^k, \mathbf{X}), \quad (\text{A-1})$$

where $\tilde{\mathbf{Y}}_s^k$ is the subset of points of \mathbf{Y}_s^k that falls within the horizontal extent of their patch $[-1, 1]^2 \times \mathbb{R}$. To prevent the slots from predicting shapes outside of the patch extent, we regularize our model by the square Euclidean distance between the output of $\mathcal{D}_{\text{translate}}$ and the set $[-1, 1]^2 \times \mathbb{R}$ for each slot.

Training. We use the efficient CUDA implementation of the Chamfer distance by PyTorch3D [10] which signifi-

cantly speeds up training. We use the ADAM optimizer [4] with a learning rate of 10^{-4} and default parameters, except for the prototypes’ intensities, scales and points’ positions which we learn without weight decay.

Curriculum learning. Following the ideas of Monnier *et al.* [7] and Loiseau *et al.* [5], we use a multi-stage curriculum strategy to prevent our model from falling in bad minima. We gradually unfreeze the model parameters in the following order: (i) translation, rotation, tilt, slot activation, and choice of prototype; (ii) intensities of the prototypes, when available; (iii) scales of the prototypes; (iv) shapes of the prototypes (positions of their 3D points); (v) anisotropic scalings of the prototypes. Alignment networks are initially set to identity by setting the parameters of the decoders’ last linear layers to zero. When unfreezing a new module, the learning rate of all the model’s parameters is set to 1/1000 of the global learning rate and gradually increased over 1000 batches to the global learning rate to smooth the training and benefit from what has been learned previously by the encoder. We define an “epoch” as 512 batches of 64 patches, and each stage of the curriculum is trained until convergence.

Table A-1. **Learnable Earth Parser hyperparameters.** Choice of hyperparameters when training on the Earth Parser Dataset. We used similar configurations across scenes, only adapting the voxel size and number of slots.

Scene	Voxel size (cm)	number of slots S
Crop Fields	40	64
Forest	60	64
Greenhouses	60	64
Marina	20	64
Power Plant	60	128
Urban	40	64
Windturbines	320	64

Scene-specific hyperparameters. We trained our model on each seven scenes of the Earth Parser Dataset, with minor adaptation shown in Table A-1. We only change the size of the voxel grid to adapt our reconstruction model to the size of the typical object we want to discover. For example, a windturbine is typically 100 meters tall, while a boat is typically 5 meters long. We also doubled the number of slots for the “Power Plant” scene because of its geometric complexity.

ShapeNet adaptations. As the objects of ShapeNet-Part [11] are simple, we only use $S = 6$ slots. To account for the diversity of the size and shapes of parts, we replaced the anisotropic-scaling transformation by an unconstrained affine transformation.

Ablation Study. The structure of our ablation study intentionally mirrors the curriculum learning. Specifically, we remove components in decreasing order of their impact on the reconstruction quality. Removing the translation while retaining all other transformations would lead to poor learning dynamics [7]. Our approach ensures that each step of the ablation progressively assesses the impact of each component.

A-2. Earth Parser Dataset Details

Classes names. As show in Table A-2, each scene of the Earth Parser Dataset is annotated with different classes among “ground”, “vegetation”, “building”, “boats”, “bridge”, “electric lines”, and “windturbine”.

Localization. We report the localization of the scenes of Earth Parser Dataset in Table A-2. Our dataset has been acquired in various environments distributed on the French territory.

A-3. Additional Quantitative Results

Results on the Earth Parser Dataset. We report in Table A-3 detailed results for the baselines and our method. We evaluated the use of elevation and LiDAR intensity for the k-means [6] baseline, the use of intensity in a way similar to ours for AtlasNet v2 [3], and the effect of our prototype selection post-processing step:

- **k-means features.** The use of both intensity and elevation gives a small boost to semantic performances. However, we see that when clustering with a small number of centroids ($K = 6$, as in our model), using only the elevation gives a reasonable baseline.

- **AtlasNet v2 intensity.** We extend AtlasNet v2 to handle intensity in a manner similar to our approach, which improves its segmentation results. However, AtlasNet v2 uses the same number of prototypes for each input regardless of its complexity and thus does not achieve high semantic segmentation scores.

- **Prototype selection post-processing.** On the Learnable Earth Parser, we see that our post-processing step has a limited impact on the quality of the prediction and reconstructions, except for the scene “Forest” for which the segmentation score goes from 87.3 to 80.5. This step can either increase (“Crop Fields”, “Forest”, “Marina” scenes) or decrease (“Windturbines” scene) the reconstruction quality. We believe this is because of the regularization loss which encourages all prototypes to be used. Finally, this simple post-processing step allows us to significantly decrease the number of prototypes and adapt it to the scene complexity.

Results on ShapeNet. Our experiment on ShapeNet is intended as a sanity check in a controlled setting. We report in Table 4 significantly better results than AtlasNet v2 [3] for planes with arbitrary orientation (+34.1 mIoU). We repeated the experiment for guitars and chairs and observed improvements of +23.1 and +1.5 mIoU, respectively.

Additional ablations. The coordinates of the prototypes’ points are directly learned as parameters of the model in an unsupervised fashion with our regularized reconstruction loss. We choose point cloud prototypes for their simplicity and expressivity. We also trained our model using cuboids or superquadrics as prototypes, by learning their parameters (xyz -scales for cuboids, and xyz -scales and α_1, α_2 for superquadrics) as free parameters of the model for each prototypes. This leads to worse reconstruction results (respectively, +69.3% and +42.5% increase in Chamfer distance) and segmentation (respectively, -15.8 and -17.4 mIoU) results on average on all scenes of the Earth Parser Dataset. While these shapes are more *compact* (fewer degrees of freedom), the associated reconstructions appear less legible and interpretable. They also fail to capture the diversity of

real-world 3D data (houses, trees, windmills, boats, etc.).

A-4. Additional Qualitative Results

Earth Parser Dataset results. We show in Figure A-2 the ground truth semantic segmentation, our predicted semantic segmentation, our reconstruction and our learned prototypes. They showcase the quality, interpretability, and diversity of use cases of our model on this dataset of aerial LiDAR scans. We also show some semantic and instance segmentation closeups in Figure A-3.



























Instance segmentation. We show in Figure A-4 a comparison of the instance segmentation produced by SuperQuadrics [9] and our Learnable Earth Parser. Since SuperQuadrics [9] uses a restricted family of 3D shapes to reconstruct an input scene, it has worst qualitative performances for instance segmentation when compared to our model, which learns scene-specific prototypes and can provide semantic information.

Generalizability. Our model trains individually per scene, taking around 12 hours each. We observed qualitatively that a model trained on one scene adapts well to other scenes of similar natures (e.g., different forests) but not otherwise. Training a universal model for diverse scenes is possible but would require significant memory due to the large number of prototypes needed.

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2
- [2] Spconv Contributors. SPConv: Spatially sparse convolution library. <https://github.com/traveller59/spconv>, 2022. 1
- [3] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3D shape generation and matching. *NeurIPS*, 2019. 3, 7
- [4] Diederik P Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. *ICLR*, 2014. 2
- [5] Romain Loiseau, Tom Monnier, Mathieu Aubry, and Loïc Landrieu. Representing shape collections with alignment-aware linear models. In *3DV*, 2021. 1, 2
- [6] J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, 1967. 3, 7
- [7] Tom Monnier, Thibault Groueix, and Mathieu Aubry. Deep Transformation-Invariant Clustering. *NeurIPS*, 2020. 2, 3
- [8] Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. In *ICCV*, 2021. 7
- [9] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3D shape parsing beyond cuboids. In *CVPR*, 2019. 4, 7, 8
- [10] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3D deep learning with PyTorch3D. *arXiv:2007.08501*, 2020. 2
- [11] Manolis Savva, Angel X. Chang, and Pat Hanrahan. Semantically-Enriched 3D Models for Common-sense Knowledge. *CVPR 2015 Workshop on Functionality, Physics, Intentionality and Causality*, 2015. 3

Table A-2. **Earth Parser Dataset classes and localisation.** We show the class names and color codes for the seven scenes of our dataset. Unlabeled points are represented in black ■. The Earth Parser Dataset was acquired at different locations in France, spanning a wide variety of environments.

 Crop Fields	 Forest	 Greenhouses	 Marina	 Power Plant	 Urban	 Windturbines
 Ground	 Ground	 Ground	 Boats	 Ground	 Ground	 Ground
 Vegetation	 Vegetation	 Vegetation	 Bridge	 Vegetation	 Vegetation	 Vegetation
		 Building		 Building	 Building	 Windturbine
				 Electric lines		



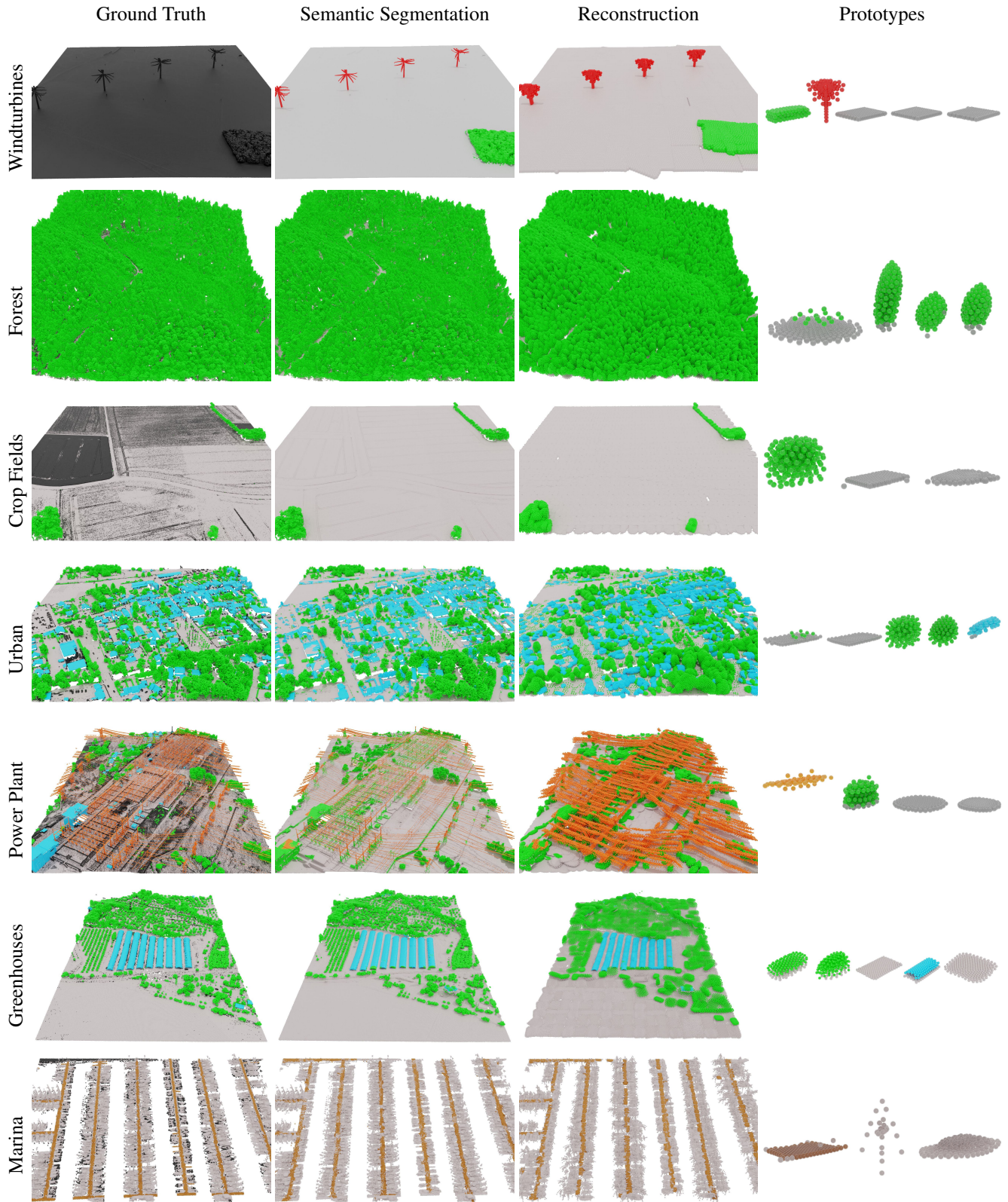


Figure A-2. **Qualitative Results.** For all scenes of the Earth Parser Dataset, we show the ground truth labels, the semantic segmentation, reconstruction, and prototypes learned by our Learnable Earth Parser.

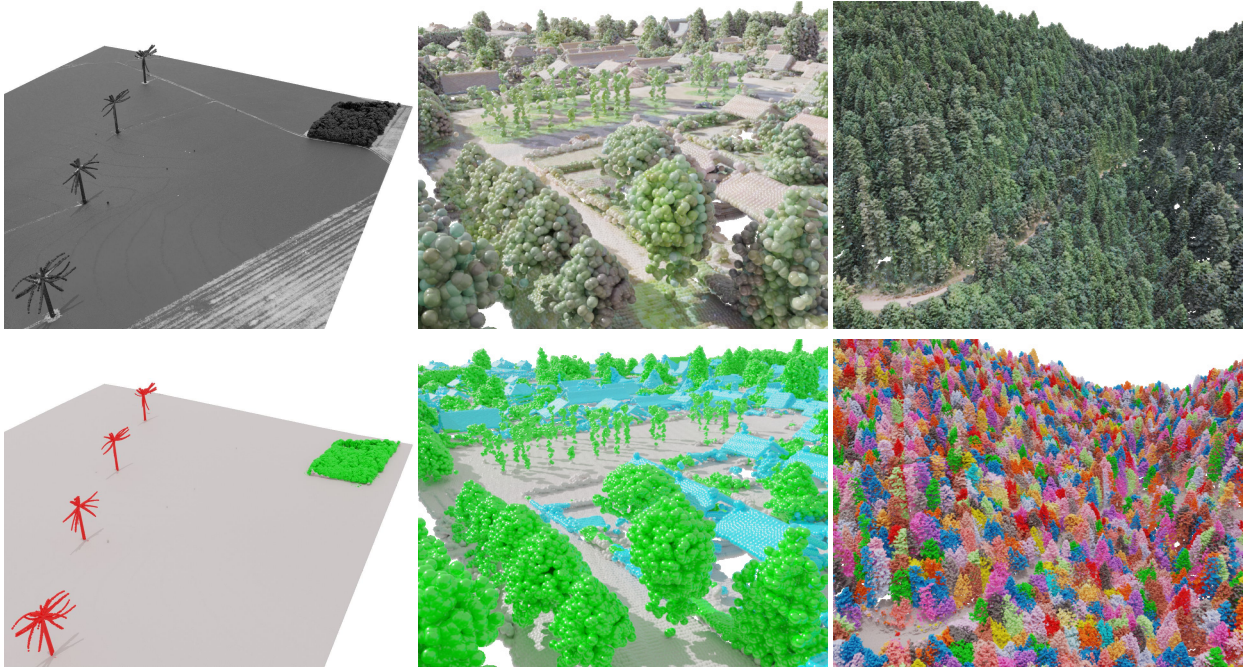


Figure A-3. **Qualitative Semantic and Instance Segmentation.** Our Learnable Earth Parser can perform semantic and instance segmentation on various scenes with minor adaptations.

Table A-3. **Results on the Earth Parser Dataset.** We report the quality of the reconstruction (Cham.) and semantic segmentation (mIoU) for the models presented in the main paper and *other variations*. **Bold** numbers indicate the best results of the models shown in the main submission, while **green bold** numbers indicate the best results across all variations. \downarrow indicates that the variation results in a significant drop in performance, while \uparrow indicates a performance boost. We also show the number of prototypes selected by our post-processing selection algorithm.

	Rec.	Semantic	Crop Fields		Forest		Greenhouses		Marina		Power Plant		Urban		Windturbines	
			Cham.	mIoU	Cham.	mIoU	Cham.	mIoU	Cham.	mIoU	Cham.	mIoU	Cham.	mIoU		
k-means (i,z) [6]	\times	\checkmark	—	93.8	—	71.5	—	39.3	—	41.4	—	42.8	—	56.5	—	87.6
k-means (i) [6]	\times	\checkmark	—	\downarrow 74.5	—	\downarrow 45.5	—	\downarrow 36.3	—	\downarrow 41.4	—	\downarrow 28.8	—	\downarrow 42.5	—	\downarrow 64.1
k-means (z) [6]	\times	\checkmark	—	93.9	—	71.4	—	39.2	—	41.4	—	42.3	—	56.2	—	\downarrow 77.5
SuperQuadratics [9]	3D	\times	0.86	—	1.04	—	0.60	—	0.93	—	0.58	—	0.40	—	13.5	—
DTI-Sprites [8]	2.5D+i	\checkmark	6.10	83.2	14.59	40.2	5.36	42.0	6.16	41.4	5.36	29.0	2.99	47.3	36.19	25.9
AtlasNet v2 [3]	3D+i	\checkmark	1.07	43.1	1.58	71.4	0.56	49.1	0.73	42.1	0.45	41.6	0.63	48.8	9.47	48.1
AtlasNet v2 [3] w/o intensity	3D	\checkmark	1.08	43.1	\downarrow 1.92	\uparrow 74.4	\uparrow 0.49	\downarrow 46.0	\downarrow 0.80	\downarrow 40.8	0.43	\downarrow 38.7	\downarrow 0.70	\downarrow 40.4	\uparrow 7.56	\downarrow 25.9
Ours	3D+i	\checkmark	0.72	96.9	0.88	83.7	0.40	91.3	0.82	78.7	0.44	52.2	0.29	83.2	6.65	93.4
Ours w/o post-processing	3D+i	\checkmark	\downarrow 1.02	96.5	\downarrow 0.97	\uparrow 88.0	0.38	90.7	\downarrow 0.96	78.3	0.42	52.4	0.28	83.7	6.59	\uparrow 96.4
Ours # of selected prototypes	—	—	3		4		5		3		4		5		5	

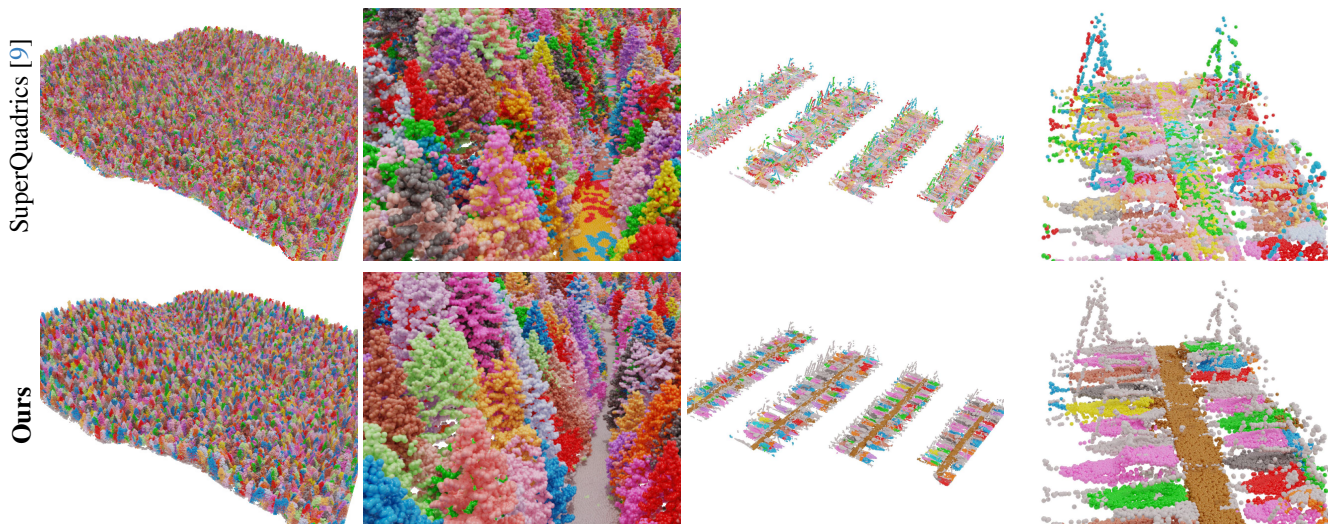


Figure A-4. **Instance Segmentation.** We represent the instances predicted with our algorithm and by SuperQuadric [9]. We see that SuperQuadrics' reconstruction struggles modeling complex objects with only one instance. Moreover, our method make it easier to differentiate between different object types such as "trees" or "boat hull", while all superquadric are generated in the same way.