2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Feature-Realistic Neural Fusion for Real-Time, Open Set Scene Understanding

Kirill Mazur, Edgar Sucar and Andrew J. Davison

***Abstract*— General scene understanding for robotics requires flexible semantic representation, so that novel objects and structures which may not have been known at training time can be identified, segmented and grouped. We present an algorithm which fuses general learned features from a standard pre-trained network into a highly efficient 3D geometric neural field representation during real-time SLAM. The fused 3D feature maps inherit the coherence of the neural field's geometry representation. This means that tiny amounts of human labelling interacting at runtime enable objects or even parts of objects to be robustly and accurately segmented in an open set manner. Project page: https://makezur.github.io/FeatureRealisticFusion/**

I. Introduction

Robots which aim towards general, long-term capabilities in complex environments such as homes must use vision and other sensors to build scene representations which are both geometric and semantic. Ideally these representations should be general purpose, enabling many types of task reasoning, while also efficient to build, update and store.

Semantic segmentation outputs from powerful single-frame neural networks can be fused into dense 3D scene reconstructions to create semantic maps. Systems such as SemanticFusion [1] have shown that this can be achieved in real-time to be useful for robotics. However, such systems only make maps of the semantic classes pre-defined in training datasets, which limits how broadly they can be used. Further, their performance in applications is often disappointing as soon as real-world conditions vary too much from their training data.

In this paper we demonstrate the advantages of an alternative real-time fusion method using general learned features, which tend to have semantic properties but remain general purpose when fused into 3D. They can then be grouped with scene-specific semantic meaning in an open-set manner at runtime via tiny amounts of labelling such as a human teaching interaction. Semantic regions, objects or even object parts can be persistently segmented in the 3D map.

In our method, input 2D RGB frames are processed by networks pre-trained on the largest image datasets available, such as ImageNet [2], to produce pixel-aligned banks of features, at the same or often lower resolution than the input frames. We employ either a classification CNN [3] or a Transformer trained in a self-supervised manner [4]. We deliberately use these off-the-shelf pre-trained networks to make the strong point that any sufficiently descriptive learned features are suitable for our approach.

{k.mazur21, e.sucar18, a.davison}@imperial.ac.uk
Dyson Robotics Lab, Imperial College London, UK. Research presented in this paper has been supported by Dyson Technology Ltd.

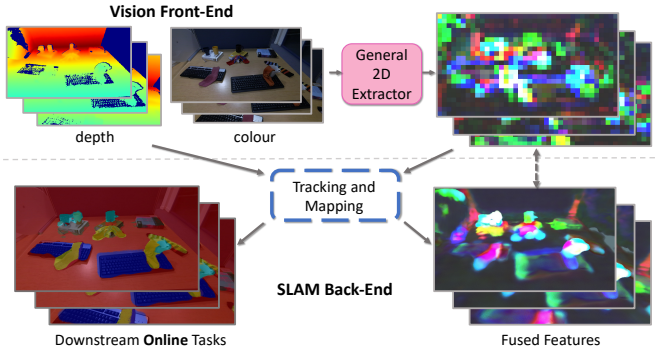

Fig. 1: **Method Overview.** We fuse general pre-trained features into a coherent 3D neural field SLAM model in real-time. The fused feature maps enable highly efficient open set scene labelling during live operation.

Rather than fusing features via essentially painting feature distributions onto an explicit 3D geometric reconstruction as is done with semantic classes in [1], here we represent geometry and feature maps jointly via a neural field. Neural fields have been recently shown to enable joint representation of geometry and semantics within a single network, such as in the off-line SemanticNeRF system [5]. The great advantage of this is that the semantic representation inherits the coherence of shape and colour reconstruction, and this means that semantic regions can accurately fit the shapes of objects even with very sparse annotation.

We base our new real-time neural feature fusion system on iMAP [6], a neural field SLAM system which uses RGB-D input to efficiently map scenes up to room scale. We augment iMAP with a new latent volumetric rendering technique, which enables fusion of very high dimensional feature maps with little computational or memory overhead.

We call our scene representation "feature-realistic" as a counterpoint to the "photo-realistic" scene models which are the aim of many neural field approaches such as NeRF [7]. We believe that robotics usually does not need scene representations which precisely model the light and colours in a scene, and that it is more valuable and efficient to store abstract feature maps which relate much more closely to semantic properties.

We demonstrate the scene understanding properties of our system via an open-set semantic segmentation task with sparse user interaction, which represents the way a human might interact with a robot to efficiently teach it about a scene's properties and objects. The user uses a few pointing

clicks to give labels to pixels, and the system then predicts these label properties for the whole scene. We show that compelling dense 3D scene semantic mapping is possible with incredible sparse teaching input at runtime, even for object categories which were never present in training datasets. Usually the user only needs to place one click on an object of a certain type for all instances of that class to be densely segmented from their surroundings. We evaluate the system on a new custom open-set video segmentation dataset.

To summarise, our contributions are as follows:

- The first neural field feature fusion system operating in *real-time*;
- A system that operates *incrementally* and successfully handles exploration of previously unobserved scene regions;
- A *latent volumetric rendering* technique which allows fusion up to **1536**-dimensional feature-maps with negligible performance overhead compared to iMAP and a scene representation of only 3 MB of parameters;
- Dynamic open set semantic segmentation application of the presented method.

II. Related work

SemanticFusion [1], an extension of ElasticFusion [8], introduced a mechanism to incrementally fuse 2D semantic label predictions from a CNN into a three-dimensional environment map. Among other similar systems, the panoptic fusion approach of [9] made an advance by explicitly representing object instances alongside semantic region classes. The latest systems in this vein wield neural fields as an underlying 3D representation. The advantageous properties of the coherence of neural fusion were first shown by Semantic NeRF [5], with variations aimed towards multi-scene generalisation and panoptic fusion demonstrated in [10, 11].

The aforementioned methods suffer from a training/runtime domain gap and the inherently closed-set nature of a fixed semantic label set. The domain is fixed by the dataset and the closed target label set employed during the semantic segmentation model pre-training. Our method relates to two recently released approaches, Distilled Feature Fields (DFF) [12] and Neural Feature Fusion Fields (N3F) [13], which also add a feature output branch to a neural field network and supervise the renders with the outputs of a pre-trained feature extractor.

Unlike our work, N3F and DFF supervise neural fields with up to 64- and 384-dimensional feature maps respectively, which is $24\times$ and $5\times$ times smaller than our proposed method. Both DFF and N3F operate in an off-line protocol similar to NeRF and require approximately a day to converge on a single scene, whereas our system operates at *interactive frame rates* making it useful for robotics. Additionally, N3F heavily leverages offline assumptions on an input sequence: all frames have to be known prior to training, due to a pre-processing step which executes dimensionality reduction jointly on all input feature maps. In our online execution paradigm these assumptions would be fundamentally violated and the input distribution might change drastically in a few seconds (e.g. entering a new room).

Both N3F and DFF mainly consider object retrieval and 3D object segmentation mask extraction scenarios. In contrast, we focus on extracting *all* object instances of varying appearance and geometry, given a semantic class. While DFF also considers the semantic segmentation scenario, it fuses the penultimate activations of a pre-trained semantic segmentation model. This method is therefore essentially equivalent to a SemanticNeRF-style approach with the same benefits and pitfalls, such as the domain gap.

Our method achieves real-time performance by using a core neural field SLAM approach based on iMAP [6], with a small MLP network, RGB-D input and guided keyframe and pixel sampling for efficiency. This type of efficient network is well suited to semantic and label fusion. Recent work iLabel [14], also based on iMAP, showed a type of interactive scene segmentation based on no prior training data. The coherence of the neural field alone was shown to be a basis for segmenting objects from sparse interaction. However, in iLabel there was little evidence that annotation of an object led to grouping with other instances of the same class. In our work we specifically show that this becomes possible due to fusion of general features from an off-the-shelf pre-trained network.

Our method also closely relates to SemanticPaint [15], an older online interactive labelling system. SemanticPaint, like our system, operates by propagating user-given labels to novel object instances. However, propagation is severely limited to objects which are almost identical apart from colour. The core of the SemanticPaint is a random forest classifier with hand-crafted features and refinement with a Conditional Random Field. This machinery cannot compete in pattern recognition abilities with the modern deep learning methods for computer vision our approach builds on. Our system benefits both the best properties of neural fields which encourage coherent segmentation [14], and the power of features from general pre-trained networks.

III. Method

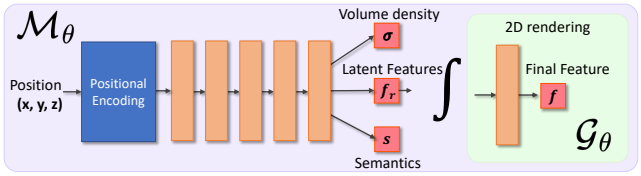

Fig. 2: **Scene Network.** Overview of our Scene Network. Our scene MLP predicts semantics and latent features, which are further refined after the volumetric rendering.

Our system is composed of two principal components: a pre-trained frozen 2D image feature extractor (vision *front-end*) and an iMAP-like [6] SLAM system (SLAM *back-end*). When our resulting system is applied for semantic segmentation, we follow iLabel [14] in extending iMAP for scene semantics prediction supervised via sparse user's annotations. See detailed discussion in the beginning of Section IV. While

our method technically allows an image feature extractor for the vision front-end of any choice, we focus on ones that are general, i.e not trained for dense prediction tasks.

Our general approach is to approximate via volumetric rendering a set of feature maps $\{\boldsymbol{F}_i = \mathcal{F}(\boldsymbol{I_i}) \in \mathbb{R}^{k \times H' \times W'}\}_i$ obtained with a feature extractor $\mathcal{F}$ from a set of images $\{\boldsymbol{I_i} \in \mathbb{R}^{3 \times H \times W}\}_i$, similar to [12, 13]. Unlike those methods, we abstain from modelling colour and view-dependent effects to ease the problem for the scene MLP. While colour may seem a more compact representation compared with a high-dimensional feature map, it inherently contains "nuisance" variation (due to e.g. illumination changes or camera settings such as auto-exposure), which is usually not relevant to semantic understanding of the scene.

A. Scene Network

The architecture of our scene mapping system is largely based on iMAP, whose notation we follow [6]. Our scene representation network $\mathcal{O}_\theta = (\mathcal{M}_\theta, \mathcal{G}_\theta)$ has two components: a NeRF-style multi-layer perceptron $\mathcal{M}_\theta$ which serves as a "scene map" and represents a three-dimensional neural field; and a single layer perceptron $\mathcal{G}_\theta$ which operates in 2D and upsamples volumetrically rendered features to the target dimension k (see Section III-B).

The coordinate MLP $\mathcal{M}_\theta$ with a hidden layer dimension $h = 256$ maps a 3D position $\boldsymbol{p} = (x, y, z) \in \mathbb{R}^3$ into $\mathcal{M}_\theta(\boldsymbol{p}) = (\rho, \boldsymbol{f}, \boldsymbol{s})$, where ρ stands for volumetric density, $\boldsymbol{f}$ for a feature vector, and $\boldsymbol{s}$ for semantic logits. Before feeding a point $\boldsymbol{p} = (x, y, z)$ into the scene network $\mathcal{M}_\theta$ we apply off-axis positional encoding [16] to ensure rich representational capacity for fitting high-frequency feature maps and mitigate axis-aligned artifacts caused by the standard positional encoding used in NeRFs.

Given N samples along a ray at depths $\{d_i\}_{i=1}^N$, we transform the corresponding densities ρ_i into ray termination probabilities $o_i = 1 - \exp(-\rho_i \delta_i)$ and further into volumetric rendering weights $w_i = o_i \prod_{j=1}^{i-1}(1 - o_j)$, where δ_i is the inter-sample distance in the volumetric integration quadrature. We render depth $\hat{D}$, features $\hat{F}_r$, and semantic logits $\hat{S}$ as:

$$\hat{D}[u,v] = \sum_{i=1}^{N} w_i d_i, \;\; \hat{F}_r[u,v] = \sum_{i=1}^{N} w_i \boldsymbol{f}_i, \;\; \hat{S}[u,v] = \sum_{i=1}^{N} w_i \boldsymbol{s}_i \tag{1}$$

We adopt iMAP's keyframing strategy: we add a frame into the keyframe set if the current depth relative error is higher than a threshold for more than 65% of pixels.

The spatial resolution of the feature maps produced by a feature extractor is usually smaller than that of original colour and depth image. To mitigate this we employ a sparse supervision technique for features, similar to the one introduced in SemanticNeRF [5], to let the scene network learn the appropriate feature spatial interpolation. To supervise depth pixels we employ random uniform sampling across the full image resolution for each mapping step.

B. Latent Feature Rendering

NeRFs are well known for their training and inference inefficiency due to the cubic complexity of volumetric rendering. This has led to an extensive line of work endeavoring to alleviate this issue, with recent highlights such as Instant NGP which adds grids and hashing to the MLP representation [17]. However, here we choose to stick with a simple MLP as our master scene representation, because of its attractive compression and coherence properties [14]

To render a single feature image of spatial dimension $H' \times W'$ a scene network has to be queried $H' \times W' \times N_{\text{bins}}$ times, where N_{bins} stands for the number of samples per ray. Therefore, given a feature map of k-dimensional features, the memory requirements scales as $k \times H' \times W' \times N_{\text{bins}}$. When the dimensionality of the target features is an order of magnitude higher than the hidden scene MLP dimension (1536 and 256 respectively in our case) the naïve approach becomes intractable, especially for a real-time system.

Our solution is to render a *latent* h-dimensional feature vector $\hat{F}_r[u,v]$ followed by a per-point perceptron $\mathcal{G}$ applied *after* the rendering:

$$\hat{F}[u,v] = \mathcal{G}_\theta \left(\sum_{i=1}^{N} w_i \boldsymbol{f}_i \right) \tag{2}$$

This simple approach enables our system to yield up to $k = 1536$ dimensional features with a negligible performance and memory overhead.

C. Feature Extractors

We have observed that models producing highly view-dependent features (such as a standard ViT [18]) are unsuitable for our application; view-equivariant effects effectively lead to an underconstrained problem for on-the-fly semantics extraction due to the incremental and online nature of our system. While most ConvNets inherit a shift invariance property from convolution and yield satisfactory feature maps, Transformer-based models tend to be highly equivariant due to the absence of inductive biases. An interesting exception is the Transformer-based DINO [4] model, pre-trained in an unsupervised setting with a learning objective to produce features invariant to a large set of image transformations.

In this work we test both convolution-based [19] and a Transformer-based [20] models to demonstrate that our method is agnostic to the nature of 2D feature front-end and is still capable of fusing these features. The feature extractor network runs inference at the same FPS as the mapping process (2 Hz, the same as iMAP) to save computation.

For our ConvNet representative model we choose EfficientNet [3], a supervised CNN trained on ImageNet [2]. Each image is passed through the pre-trained network in its original resolution and the output of the final convolutional layer is taken as the target for our system. This process yields a coarse feature map of spatial dimension 22×38 and each pixel contains a $k = 1536$-dimensional feature vector. To mitigate the artifacts caused by padding [21] we ignore a 1×2-pixel wide frame.

Our Transformer-based model is DINO pre-trained on the ImageNet corpus in a self-supervised manner. We employ the smallest model variant with output feature dimension $k =$

384 to achieve real-time performance and fit into the GPU memory. An image is dissected onto patches of size 16×16 and then fed through the Transformer network.

Our mapping network feature branch is supervised with an $L2$ distance loss in both cases.

IV. Experiments

Given a real-time RGB-D video stream $\{(\mathbf{I_i}, \mathbf{D_i})\}$, our system gradually fuses the incoming feature maps $\{\boldsymbol{F}_i = \mathcal{F}(\boldsymbol{I}_i)\}$ from the front-end and tracks itself in the scene using the SLAM back-end. A user then gradually introduces new classes with "clicks" to label *single* pixels in one of the system's keyframes. These labels are used to supervise the semantic head of the scene network $\mathcal{O}_\theta$, resulting in a 3D semantic segmentation field, similar to SemanticNeRF [5]. Inspired by iLabel [14], we use interactive labeling as a way to define "tasks" for our system on the fly. This lightweight interaction is representative of how a human might interact with a running robot system to efficiently teach it about named scene properties; or it could represent experimental scene interactions that a robot could carry out for itself. Note that our system operates in a *one-click-per-class* mode: every click defines a *new* semantic class. This is unlike iLabel, where several clicks are usually needed to identify large objects or multiple instances of a class.

The system therefore is expected to propagate the semantic label of a click across relevant scene regions, e.g. multiple instances of the same object or object parts. Since we stick to a one click-per-class execution protocol, our system is not a labelling system but rather a method to reveal the already present scene part similarities.

We qualitatively evaluate our system in three experiments:

- **Coverage:** Grouping objects in rare class scenarios (a sock, a trainer, or a GPU), where traditional models require re-training for a novel class distribution;
- **Specialization**: The ability of our system to specialize from a holistic object category into object part categories, e.g. from a mug category into two separate mug handle and body classes;
- **Exploration**: Given a labelled and reconstructed part of the scene, how well labels propagate to previously unreconstructed regions.

We strongly encourage watching our accompanying **qualitative demo video**. We also quantitatively evaluate the semantic segmentation quality of our method against the baselines on our tabletop dataset.

In most experiments we employ the EfficientNet CNN feature front-end unless stated otherwise. We chose the CNN front-end due its ability to provide stronger semantic entanglement compared to its DINO counterpart; see Section IV-E.

A. Dataset

Due to the absence of available RGB-D tabletop video datasets which contain repetitive semantic objects, particularly from the unusual classes where the performance of our method is especially notable, we chose to collect our own dataset. It consists of 8 sequences in total and captured with a handheld Microsoft Azure Kinect camera. The dataset incorporates common household and office objects, such as books, keyboards, trainers, socks, and plants as well rare objects, such as gamesticks and GPUs.

We randomly sampled $\sim 5\%$ of the frames per video sequence for five sequences (the sequences from Figure 3 and Figure 6), and then densely annotated them with ground-truth labels to obtain quantitative results in Table I.

B. Coverage

The first set of experiments focuses on evaluating semantic class coverage. In other words, to what extent is our method able to propagate semantic labels from one object to other instances of the same class. The object semantic categories present in our testing scenes are typically not covered by off-the-shelf models, such as a pre-trained Mask R-CNN [22]. Furthermore, most of these classes are not present as a target class in the ImageNet dataset, on which our EfficientNet CNN was pre-trained.

Most sequences we use are captured such that the first frame contains the whole scene. We choose this approach to ease the qualitative evaluation, so that all target objects are visible. Our system is also conceptually capable of capturing inherently 3D scenes, which do not fit into a single frame. We additionally cover such cases in Section IV-D.

In Figure 3 we demonstrate our method's performance after executing it on four tabletop sequences. We observe that our method is capable of propagating semantic class labels across a variety of objects and produces plausible semantic object masks. Note that the method extracts these masks despite fusing very coarse initial CNN feature maps of spatial dimension 22×39.

These experiments show that our method particularly shines on unusual object categories, which are typically not present in traditional densely annotated datasets.

C. Specialization

Another potential benefit of our approach not being restricted to a fixed class set is the ability to split classes further down the natural semantic hierarchy. To test out this property we captured scenes with several instances of semantically composite objects, mugs and headphones. We also equip the scenes with an additional object of unrelated class (book) to ensure that the observed label propagation is not incidental.

First, as in the previous experiment we assign one label to the target object instance, one to the background, and one to the unrelated object. In this initial step we observe similar behaviour to before: our system groups the target objects together, while separating out the background and the additional book class.

In the second stage a new label is introduced, denoting a subclass of the initial class: handle and ear cover for the base mug and headphones classes respectively. The results are illustrated in Figure 5. We observe that the system successfully dissects one object class into two and propagates this dissection to all object instances of the same class, indicating useful part or affordance representation capabilities.

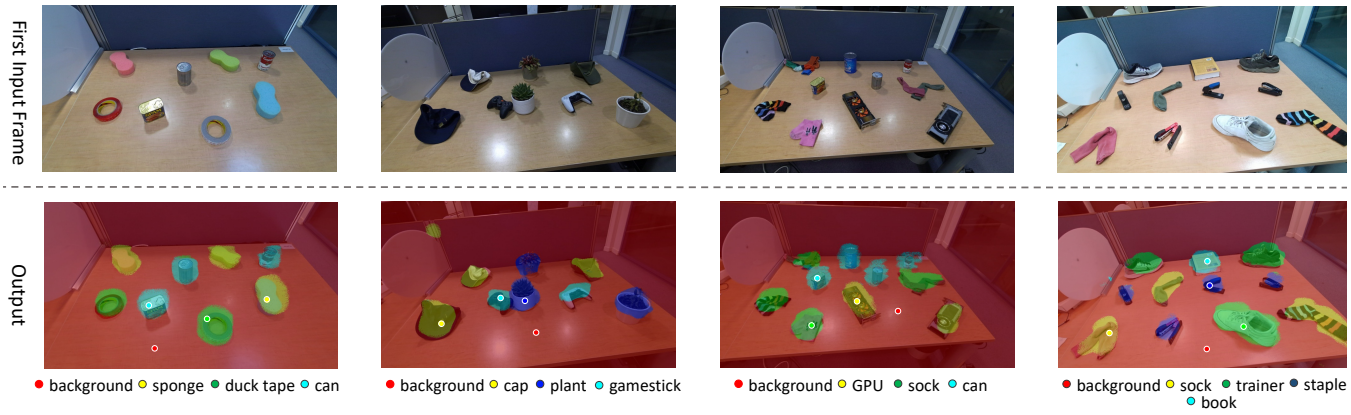

Fig. 3: **Coverage experiments.** Each RGB frame is the first of an RGB-D sequence from which we reconstruct the 3D scene and fuse features in real-time. The coloured dots in the output view are the *only* semantic annotations supplied, as clicks on the first frame. Dense semantic predictions are shown, showing high quality semantic segmentation and grouping within instance classes. Note that these results use a *very* coarse 22×38 CNN feature map.

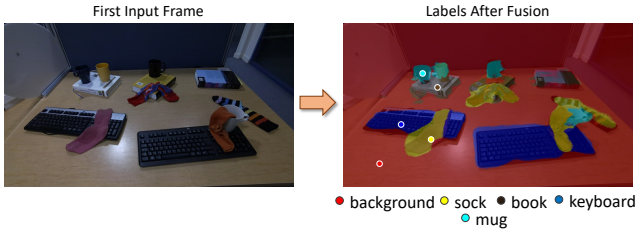

Fig. 4: **Cluttered scene.** Our method successfully handles even cluttered scenes with objects which are in contact and occluding each other.

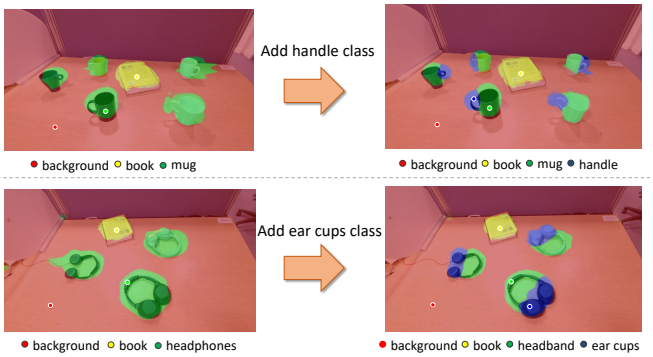

Fig. 5: **Specialization.** In these experiments we show that if one click is added to an object to indicate a new sub-part class, all instances are correctly divided into the same part classes.

D. Exploration

We also test in an exploratory setting where only part of the whole scene is visible and reconstructed at the label assignment stage: see Figure 6. First, labels are assigned in the first view to one object per class as in the experiments of Section IV-B. Then, the camera moves to view new parts of the scene with other objects. The scene network adds new keyframes automatically and continues to train its geometry/feature representation in real-time. In Figure 6 we can observe how the correct semantic labels "emerge" automatically on newly observed objects as they are reconstructed. Our method strongly propagates labels to object instances unseen at the label assignment stage: mugs, books, keyboard, monitor bases, and unobserved table regions.

E. Feature Extractor

Despite the emerging trend [23, 24] of leveraging DINO for unsupervised semantic segmentation, we have observed both qualitatively and quantitatively that a supervised CNN yields stronger results in our setting.

Our qualitative (see Figure 7) and quantitative evaluations indicate an interesting trade off between these two front-ends: a Transformer-based feature extractor provides cleaner semantic boundaries for objects, whereas a CNN-based facilitates stronger semantic entanglement for complex objects at the expense of geometric accuracy. We would expect that if the SLAM reconstruction of our system was more precise, the CNN front-end would also be able to get similarly accurate semantic boundaries.

F. Quantitative Evaluation

To evaluate the performance of our method quantitatively, we adopt a standard Mean Intersection over Union (mIoU) to measure the semantic segmentation quality of our method. We annotate some image regions as an ignore class (desk separators, single instance clutter, and other regions outside the tabletop) to only measure how well does our method handles semantic object grouping.

Since our method is complementary to and disentangled from front-end feature quality, we compare it with an unfused, purely 2D feature-based, method. In our protocol the target classes are defined via a single click, i.e a labelled pixel. Therefore, we devise a baseline which measures feature similarity between the features at a target pixel and features associated with the labelled pixels.

Recent work [23] has demonstrated the strong performance of clustering DINO features for semantic segmentation. Inspired by this approach we design our baselines using feature-metric clustering. Let $\mathbf{I_0}$ be the starting image from a test sequence with a set of user-defined clicks $\{(\zeta_i, c_i)\}$ on it, where $\zeta_i \in [0,1]^2$ is the spatial position of a pixel click and c_i is its class label. We query $\mathcal{F}(\mathbf{I_0})$ using bilinear interpolation at spatial positions ζ_i to obtain anchor features $f_i = \mathcal{F}(\mathbf{I_0})[\zeta_i]$. To classify pixels in a target image $\mathbf{I_j}$, we

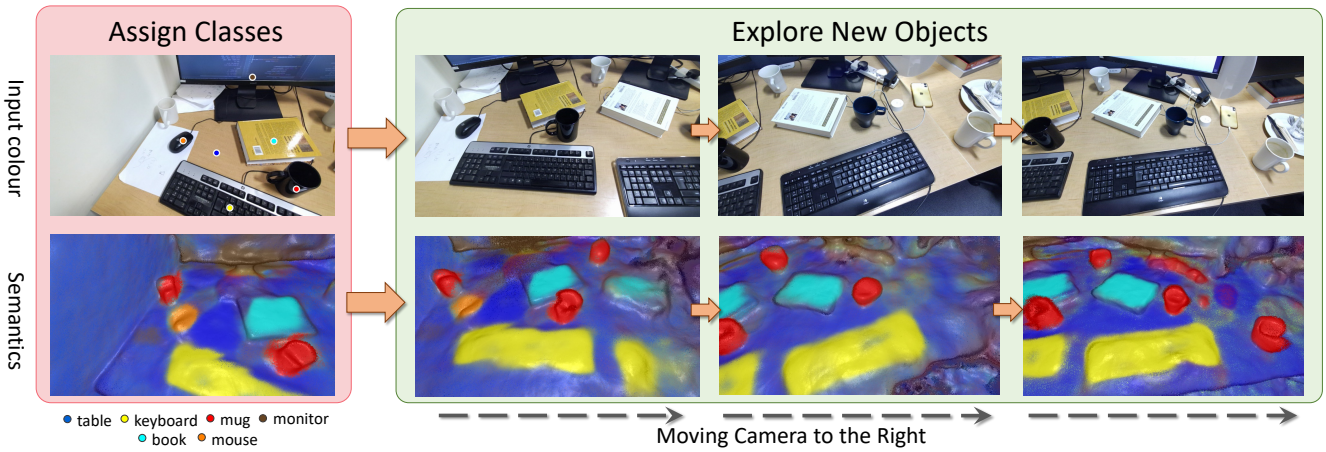

Fig. 6: **Exploration.** Objects are annotated with one click each in the first frame; dense segmentation then correctly propagates to new instances as the camera explores and the network representing reconstruction and features trains continuously.

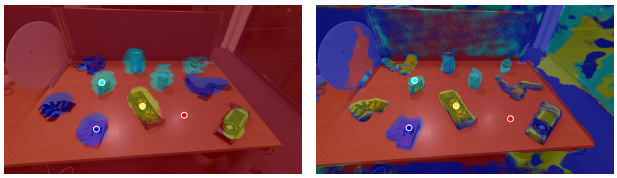

Fig. 7: **Feature Front-End Choice**. While the DINO-based feature front-end yields more geometrically-accurate masks, they have weaker semantic entanglement and some objects are ambiguously labelled.

pass it through the feature extractor $\mathcal{F}(\mathbf{I_j})$ and assign a class label in a one-nearest-neighbour classification manner, for each pixel based on the feature-metric distance to the anchor set $\{f_i\}_i$. Cosine similarity performed best in our case, which is consistent with the literature.

Furthermore we show quantitatively the benefits of our introduction of learned priors. We therefore compare the performance with that of iLabel using the same clicks. Unlike our system, iLabel does not use any prior learned information and relies purely on colour and geometry self-similarities in the underlying neural scene model.

Our method improves over the pure feature-based counterpart by a large margin for both EfficentNet and DINO front-ends. We argue that this is due to its noise-filtering and spatial upsampling properties. Interestingly, the DINO front-end outperforms EfficentNet on the desk sequence, with more common (mugs, keyboards, books) classes. This due to the fact that the DINO-based front-end provides weaker semantic entanglement, yet produces finer semantic masks, as has already been discused in Section IV-E. Meanwhile EfficentNet thrives in settings with less common objects, such as GPUs, socks, plants, etc.

G. Limitations

We provide a qualitative example in Figure 8 (left) of where our method's pixel accuracy in a cluttered scene severely degrades (yet the label assignment remains valid) due to the presence of rare classes. Meanwhile in Figure 8 (right) our method struggles to differentiate mugs with a wide

Method	Plants	Desk	Sponge	Trainers	GPUs	Mean
Fused EffcientNet (ours)	**65.1**	57.8	**59.9**	**65.4**	**59.4**	**61.5**
Fused DINO (ours)	46.4	**63.3**	56.6	42.1	52.5	52.2
EffiecentNet baseline	41.3	50.0	38.1	47.5	51.6	43.7
DINO baseline	40.4	59.1	55.4	45.4	39.4	45.9
iLabel	46.4	30.6	32.9	37.9	27.1	35.0

TABLE I: **mIOU scores.** Quantitative evaluation of our Feature-Realistic Fusion system performance on our tabeltop sequences. Feature-Realistic Fusion demonstrates consistent improvement over pure feature-based baselines for both vision front-ends as well as over the colour-based iLabel system.

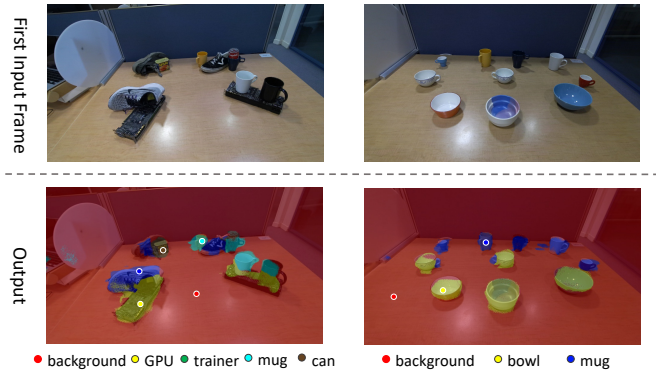

Fig. 8: **Limitations.** On the left, segmentation accuracy is degraded in a highly complex scene with unusual compound objects such as GPUs and shoes; on the right, extremely similar classes such as mugs and bowls can be confused.

body from bowls due to their natural semantic connection.

V. Conclusion

We have shown that real-time fusion of general high-dimensional features can be efficiently and simply achieved within a neural field SLAM system, and that this enables scenes to be densely semantically segmented with only a tiny amount of run-time, open-set annotation. This approach is particularly promising for robotics in complex and unusual domains where pre-trained semantic segmentation networks currently perform poorly, and we plan to soon run it at larger scale and with more complex scenes.

REFERENCES

[1] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[3] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2019.

[4] M. Caron, H. Touvron, I. Misra, H. J'egou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[5] S. Zhi, T. Laidlow, S. Leutenegger, and A. J. Davison, "In-place scene labelling and understanding with implicit scene representation," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[6] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "iMAP: Implicit mapping and positioning in real-time," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[8] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *Proceedings of Robotics: Science and Systems (RSS)*, 2015.

[9] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "Panopticfusion: Online volumetric semantic mapping at the level of stuff and things," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[10] X. Fu, S. Zhang, T. Chen, Y. Lu, L. Zhu, X. Zhou, A. Geiger, and Y. Liao, "Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation," in *International Conference on 3D Vision (3DV)*, 2022.

[11] A. Kundu, K. Genova, X. Yin, A. Fathi, C. Pantofaru, L. Guibas, A. Tagliasacchi, F. Dellaert, and T. Funkhouser, "Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] S. Kobayashi, E. Matsumoto, and V. Sitzmann, "Decomposing nerf for editing via feature field distillation," in *Advances in Neural Information Processing Systems*, 2022.

[13] V. Tschernezki, I. Laina, D. Larlus, and A. Vedaldi, "Neural feature fusion fields: 3D distillation of self-supervised 2D image representations," in *Proc. of Joint 3DIM/3DPVT Conference (3DV)*, 2022.

[14] S. Zhi, E. Sucar, A. Mouton, I. Haughton, T. Laidlow, and A. J. Davison, "iLabel: Interactive neural scene labelling," in *IEEE Robotics and Automation Letters*, 2023.

[15] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Nießner, A. Criminisi, S. Izadi, and P. Torr, "Semanticpaint: Interactive 3d labeling and learning at your fingertips," *ACM Transactions on Graphics*, vol. 34, no. 5, November 2015.

[16] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," 2022.

[17] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," in *Proceedings of SIGGRAPH*, 2022.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[19] Y. LeCun and Y. Bengio, *Convolutional Networks for Images, Speech, and Time Series*, 1998.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems (NIPS)*, 2017.

[21] B. Alsallakh, N. Kokhlikyan, V. Miglani, J. Yuan, and O. Reblitz-Richardson, "Mind the pad – cnns can develop blind spots," in *International Conference on Learning Representations*, 2021.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[23] L. Melas-Kyriazi, C. Rupprecht, I. Laina, and A. Vedaldi, "Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization," in *CVPR*, 2022.

[24] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep vit features as dense visual descriptors," *European Conference on Computer Vision Workshop (ECCVW)*, 2022.