University of Milan

Faculty of Science and Technology

Department of Computer Science
"Giovanni Degli Antoni"

Bachelor's Degree in Computer Science

Refinement of Pseudo-Labels in Unsupervised Domain Adaptation for Semantic Segmentation Using SAM

Supervisor: Prof. Nicola Basilico
Co-supervisor: Michele Antonazzi
Candidate: Lorenzo Signorelli
Student ID: 10135A

Academic Year 2024/2025

Contents

1. Introduction

Semantic segmentation is a key task in computer vision, essential for applications such as autonomous driving, urban scene understanding, and robotic perception. The typical workflow involves training a deep neural network (DNN) on a large annotated dataset in a controlled setting, and then deploying it in real-world environments. However, this transition often reveals a critical limitation: the distribution of input data in deployment can differ significantly from that of the training set—a phenomenon known as *domain shift*. This is not a limitation of semantic segmentation per se, but rather of current state-of-the-art DNN-based approaches to the problem. Domain shift can be caused by changes in lighting, sensor types, weather conditions, or scene layout, and frequently leads to a notable degradation in segmentation performance when the model is applied to a new domain.

To address this challenge, Unsupervised Domain Adaptation (UDA) has emerged as a field of reserch. UCDA enables a model to adapt continuously to new, unlabeled target domains without requiring any annotated data, thereby avoiding the need to retrain the model with manually labeled data, which would significantly increase costs. This makes it particularly suitable for robots that need to adapt to environments different from those they were originally trained in, such as navigating various homes with differing layouts and lighting, and visual features.

However, a critical limitation of many current UCDA methods lies in their reliance on pseudo-labels—labels generated by the model itself on the target domain, used as proxies for ground truth during adaptation. These pseudo-labels are often noisy and incorrect, especially in the early stages when the model's performance on the target domain is still limited. Even when they help improve predictions, their quality and reliability tend to be limited and irregular. As a result, errors in pseudo-labels can

propagate through the adaptation process, leading to degraded model performance and ultimately limiting the effectiveness of adaptation.

Recent advances in foundation models have transformed the landscape of computer vision by enabling general-purpose architectures capable of solving diverse tasks with minimal supervision. These models, trained on massive datasets with broad coverage, can generalize to new domains or tasks with little or no additional fine-tuning—a capability referred to as *zero-shot* or *few-shot* learning. In robotics and other real-world applications, where manual annotation is often infeasible and conditions change dynamically, such models offer significant advantages in adaptability and scalability. One such model is the Segment Anything Model (SAM) [7], which brings this paradigm to the task of image segmentation.

This thesis addresses the problem of poor pseudo-label quality in the UCDA pipeline by introducing a refinement mechanism based on the Segment Anything Model (SAM2), a powerful, general-purpose segmentation tool designed to produce high-quality masks from prompt-based inputs [10]. Our work builds upon the framework presented in *Continual Adaptation of Semantic Segmentation Using Complementary 2D–3D Data Representations* [5], which uses 3D semantic maps to generate pseudo-labels.

More specifically, the UCDA pipeline described in this work [5] generates pseudo-labels that tend to be blocky and do not fully conform to object shapes, limiting their accuracy and adaptability. Our approach leverages SAM to refine these 3D-based pseudo-labels, improving their quality and enhancing the overall adaptation performance.

We propose two strategies to leverage SAM for refining pseudo-labels within the UCDA pipeline. One uses information derived from the model's initial predictions to guide SAM's segmentation, while the other employs a spatial prompting approach to generate dense and unbiased region proposals. Both strategies are integrated into a continual adaptation framework built on the DeepLabV3 semantic segmentation architecture.

During each iteration of the adaptation loop, the model's predictions are refined using SAM, and the resulting masks are used to update the segmentation model. This

iterative process aims to gradually improve the model's performance by continuously feeding it cleaner and more reliable training signals in the form of refined pseudo-labels.

The objective of this thesis is to evaluate whether incorporating SAM-based refinement (zero-shot) improves the segmentation performance of UCDA systems in challenging target domains. Specifically, we test our approach on ScanNet [4] scenes (indoor environments) numbered 0 to 9, which were not part of the original training data for DeepLabV3 [3]. We focus on assessing the spatial accuracy and semantic consistency of the refined pseudo-labels and their impact on the overall continual adaptation process.

The structure of this thesis is as follows. Chapter 2 presents a background overview of domain adaptation, pseudo-labeling, and segmentation methods. Chapter 3 introduces the methodology used, detailing the integration of SAM into the Kimera–DeepLabV3 pipeline and describing the refinement strategies. Chapter 4 discusses the experimental setup, datasets used, and evaluation metrics, followed by an analysis of the results. Chapter 5 concludes the thesis by summarizing the key findings and outlining potential directions for future work.

2. State of the Art

Semantic segmentation models have achieved impressive performance in controlled environments, but they often struggle in real-world applications due to domain shifts between training and deployment scenarios. This chapter reviews key techniques developed to address these challenges, focusing on domain adaptation, continual learning, 2D–3D data integration, and segmentation refinement using advanced tools such as the Segment Anything Model (SAM).

2.1. Semantic Segmentation Challenges

Semantic segmentation is the task of assigning a semantic label to each pixel in an image, thereby providing a dense and fine-grained understanding of scene content. This differs significantly from object detection, which typically involves predicting bounding boxes and class labels for discrete objects. Unlike detection, segmentation must resolve detailed object boundaries and handle overlapping or occluded regions with high spatial precision. These requirements make semantic segmentation particularly sensitive to domain variations, such as changes in lighting, scene structure, and sensor characteristics.

To address these challenges, DeepLabV3 has emerged as one of the most effective architectures for semantic segmentation. It introduces the concept of Atrous Spatial Pyramid Pooling (ASPP), which applies multiple parallel atrous (dilated) convolutions at different rates to capture context at various spatial scales. This design allows the model to better handle objects of different sizes and achieve more accurate boundary delineation. DeepLabV3 is widely used as a backbone model in domain adaptation

studies due to its robust performance and flexibility [3].

2.2. Unsupervised Domain Adaptation

To bridge the performance gap caused by domain shift, the field of Unsupervised Domain Adaptation (UDA) has developed two primary approaches: self-training and domain adversarial learning.

In self-training, the model is first applied to the unlabeled target domain to generate pseudo-labels, which are then used as supervisory signals for further training. This iterative process allows the model to gradually adapt to the new domain. Class-Balanced Self-Training (CBST) is one notable method in this category; it mitigates class imbalance by prioritizing underrepresented categories and integrates spatial priors to refine label quality [15]. Other techniques, like Cycle Self-Training, employ bidirectional learning and entropy regularization to increase robustness against noisy pseudo-labels [1].

Domain adversarial methods focus on aligning the feature distributions between the source and target domains. This is typically achieved using a domain discriminator that tries to distinguish features from different domains, while the segmentation model learns to produce domain-invariant representations by deceiving the discriminator [13].

When these adaptation techniques are applied in an iterative and sequential manner across multiple target domains, without revisiting the original source data, the process is referred to as Unsupervised Continual Domain Adaptation (UCDA). UCDA addresses the challenges of adapting to evolving target environments while retaining knowledge from previously adapted domains.

2.3. Continual Semantic Adaptation

In many deployment scenarios, semantic segmentation models encounter a stream of evolving domains. Continual Unsupervised Domain Adaptation (UCDA) addresses this by enabling adaptation to new domains while preserving performance on previously seen ones—avoiding *catastrophic forgetting*[1].

Three prominent UCDA approaches are:

1. Memory-based adversarial learning. *Continual Unsupervised Domain Adaptation for Semantic Segmentation* by Kim et al. [6] uses an Expanding Target-specific Memory (ETM) that stores representative features from past domains. A Double Hinge Adversarial (DHA) loss aligns incoming target-domain features with memory content, allowing sequential adaptation (e.g., GTA5 $\rightarrow$ Cityscapes $\rightarrow$ IDD) without retaining source data.

2. Source-free feature distribution alignment. *Source-Free Continual Semantic Segmentation via Latent Distribution Alignment* by Stan and Rostami [12] models source-domain features as a Gaussian Mixture Model (GMM). Incoming unlabeled data are aligned to this distribution at deployment time, enabling adaptation without access to source samples—ideal for privacy-sensitive or storage-constrained settings.

3. Pseudo-label refinement via 2D–3D fusion. *Continual Adaptation of Semantic Segmentation using Complementary 2D–3D Data Representations* by Frey et al. [5] integrates 2D predictions into a volumetric 3D map via probabilistic accumulation. The 3D map is re-rendered as multiview 2D pseudo-labels for retraining, with an experience-replay mechanism to prevent forgetting. Evaluations on ScanNet scenes demonstrate improved segmentation (~9.9% mIoU gain) and knowledge retention.

[1]The loss of knowledge from previous domains.

2.4. 2D–3D Integration: The Role of Kimera

A key advancement in semantic segmentation is the integration of 2D image predictions with 3D spatial context to improve pseudo-label quality. Kimera, introduced by Rosinol et al. [11], is a real-time 3D reconstruction and semantic mapping system that enables this integration. It combines visual-inertial odometry, mesh reconstruction, and semantic inference to build accurate, globally consistent 3D maps of the environment.

Kimera-Semantics, one of its modules, takes 2D semantic segmentation outputs and projects them into 3D space using ray casting. As multiple views are accumulated over time, the system fuses these predictions to form a coherent and spatially accurate 3D semantic map. This process reduces noise and compensates for the limitations of single-frame 2D predictions.

Crucially, the resulting 3D semantic map can be re-projected back into the image plane, providing improved pseudo-labels that maintain spatial consistency across different frames. These reprojected labels are particularly useful for self-supervised and semi-supervised learning settings, where ground truth is limited or unavailable.

By incorporating geometric information, Kimera bridges the gap between appearance-based 2D segmentation and spatial reasoning, producing pseudo-labels that are more reliable and better aligned with the structure of the environment.

2.5. Refining Pseudo-Labels with the Segment Anything Model (SAM)

The Segment Anything Model (SAM), developed by Meta AI [7], introduces a zero-shot segmentation approach that generalizes across diverse image types without task-specific training. SAM generates high-quality masks based on simple prompts such as points, bounding boxes, or automatically sampled cues.

Unlike traditional models, SAM is instance-aware and does not require class labels, making it especially valuable for refining noisy or imprecise pseudo-labels generated in domain adaptation pipelines. It excels at delineating object boundaries with precision, which is crucial when the semantic content of the label is roughly correct but its spatial extent is inaccurate.

In this thesis, SAM is used to enhance pseudo-labels by preserving their semantic identity while improving their spatial accuracy. The refined masks, derived from SAM's prompt-based segmentation, help correct coarse or poor boundary alignment predictions around object edges. This might result in cleaner training targets and more stable self-training cycles.

By combining the label information from pseudo-labels with the fine-grained geometry of SAM masks, we improve the quality of supervision, ultimately leading to better segmentation performance in continual adaptation settings.

Together, these techniques and tools form the foundation of this thesis's approach to improving continual domain adaptation for semantic segmentation through SAM-based pseudo-label refinement.

3. Method

3.1. Overview of the Baseline Framework

This work builds upon the continual semantic adaptation pipeline introduced by Frey et al. [5], where semantic knowledge from 2D image frames is fused into a 3D geometric representation to enable pseudo-label generation for continual training. The key idea is to use a SLAM-based system[1] to build a voxel-based semantic map from RGB-D input and then project this 3D information back into the image domain via ray casting.

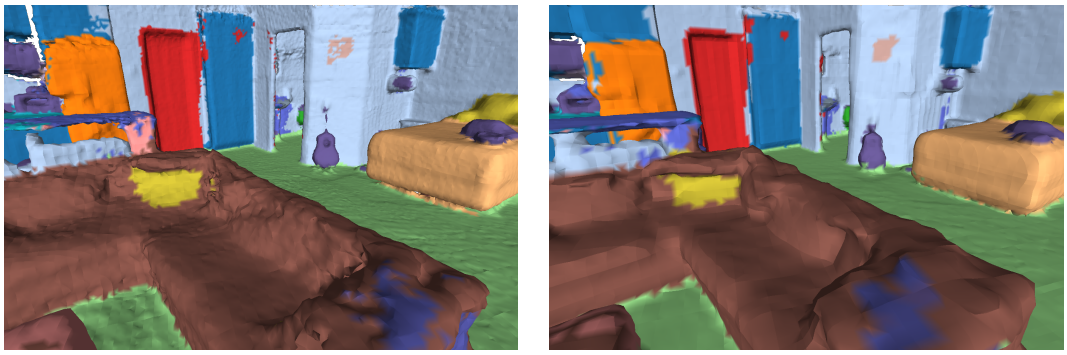
The process begins with semantic predictions obtained from a DeepLabV3 network applied to RGB images. These predictions, along with the RGB image and corresponding depth map, are passed to Kimera [11]. Kimera incrementally constructs a 3D voxel map where each voxel stores a semantic probability distribution, updated over time through Bayesian fusion. This ensures global consistency and robustness to single-frame errors.

Pseudo-labels are then generated by projecting the semantic voxel map back into the image plane using ray casting. These pseudo-labels capture the aggregated and temporally consistent semantics of the scene and are used to supervise continual fine-tuning of the segmentation model.

[1] SLAM (Simultaneous Localization and Mapping) refers to the process of concurrently estimating an agent's position and constructing a map of the environment. For a survey, see Cadena et al. [2].

In this thesis, we reproduce the method of Frey et al. [5] and use the model checkpoints and training configurations from Liu et al. [8], who also adopt the Kimera-based mapping pipeline. This provides a solid baseline from which our contributions can be evaluated.

An example of a semantic voxel map generated by Kimera is shown in Figure 3.1. The color-coded voxels reflect the semantic categories estimated through fusion of 2D predictions and 3D geometry.



Semantic Voxel Map - 3 cm | Semantic Voxel Map - 5 cm

Figure 3.1: Comparison of semantic voxel maps built by Kimera using different voxel sizes on Scene 0. A 3 cm voxel grid preserves object detail more accurately, while a 5 cm grid results in coarser, less precise structures.

All architectural and implementation details—such as network layers, resolution settings, and voxel size configuration—are provided in the next chapter.

3.2. Motivation for the Extension

Despite the effectiveness of the Kimera-generated pseudo-labels, they are susceptible to several limitations. When voxel sizes are increased (e.g., from 3 cm to 5 cm) to reduce memory and computation, spatial precision drops significantly. Boundaries between objects become blurred, thin structures may disappear, and noisy labels can emerge in areas with inconsistent depth data or occlusions.

This degradation is particularly evident when the camera is close to surfaces in the environment. At such short distances, a 5 cm voxel size becomes too coarse to capture fine-grained structural and semantic details. Larger voxel sizes introduce

semantic imprecision, especially near object boundaries and in cluttered regions. The voxel grid fails to preserve small or thin objects, and object boundaries become less distinct. In high-resolution spatial reconstructions, a smaller voxel size such as 3 cm offers better fidelity. Empirically, we observe that across evaluated scenes, using a 3 cm voxel size yields an average improvement of +1.84 mIoU over the 5 cm configuration 4.1.

Figure 3.2 demonstrates this issue using representative frames from Scene 0. Each frame is rendered twice: once using a voxel size of 3 cm and once at 5 cm. The degradation in label quality at larger voxel sizes is clearly visible.

Comparison of 3 cm vs. 5 cm Voxel Size Projections (Scene 0)

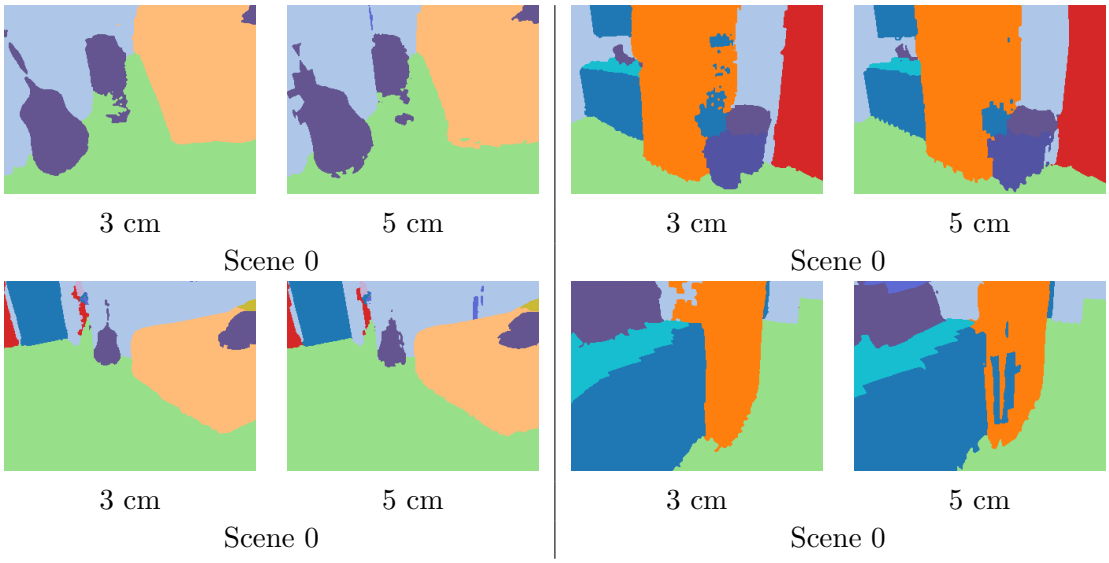


Figure 3.2: Effect of voxel resolution on pseudo-label quality. Each row shows a confrontation between 3 cm and 5 cm voxel projections from Scene 0. Larger voxel sizes introduce semantic imprecision, especially near object boundaries and in cluttered regions.

To address this limitation, we propose integrating the Segment Anything Model (SAM) [7] specifically the SAM2 version [10] into the pseudo-label generation pipeline. SAM is a powerful vision foundation model capable of producing high-resolution segmentation masks from RGB images with minimal prompting.

Our approach uses SAM to refine the pseudo-labels obtained from the Kimera voxel map before they are used for training. SAM is automatically prompted (see next section) to generate segmentation masks that are then spatially aligned with the

pseudo-labels. These masks preserve fine object boundaries and structural details, compensating for the coarseness introduced by voxel-based label projection.

Figure 3.3 illustrates this refinement process. Each row shows a comparison between the original pseudo-label from Kimera and the corresponding SAM-refined label for the same scene. The examples are taken from Scenes 7, 3, 5, and 9, demonstrating SAM's consistent improvement across diverse environments.

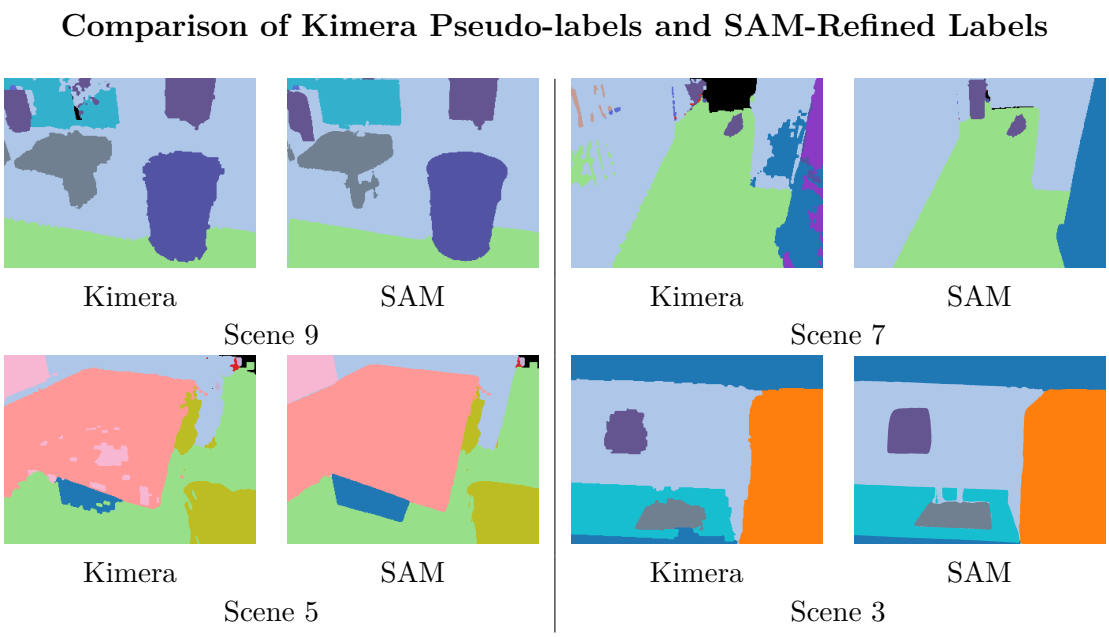


Figure 3.3: Comparison between pseudo-labels from Kimera and refined labels produced with SAM. SAM provides sharper instance boundaries and better object separation.

3.3. Proposed Architecture

The proposed architecture extends the baseline pipeline by introducing a refinement mechanism that integrates both 3D spatial information and 2D semantic accuracy. The goal is to improve the quality of the pseudo-labels used for continual training by combining the semantic context provided by Kimera with the boundary precision of the Segment Anything Model (SAM). The full pipeline consists of six key stages:

1. **Labeling with DeepLabV3:** For each RGB frame, semantic segmentation is performed using a DeepLabV3 model trained on the source domain. These predictions are used as input for the mapping process.

2. **Semantic Map Generation with Kimera:** The RGB-D input and corresponding semantic labels are passed to Kimera, which incrementally builds a 3D voxel map with probabilistic semantic annotations using Bayesian fusion.

3. **Ray Casting for Pseudo-Label Projection:** The semantic voxel map is projected back into the image plane via ray casting. This process generates dense pseudo-labels aligned with each RGB frame.

4. **RGB Segmentation via SAM Prompting:** Each RGB image is processed by SAM. Prompts are generated automatically based on scene geometry (see next section 3.4), and SAM produces high-resolution segmentation masks.

5. **Mask Fusion:** The SAM-generated masks are aligned with the Kimera pseudo-labels. In overlapping regions, class labels are transferred from the pseudo-labels to the SAM masks. Priority is given to the SAM masks in ambiguous or low-confidence areas to improve boundary accuracy.

6. **Continual Training:** The refined masks are used as supervision for continual training of the DeepLabV3 model. This enables adaptation to new domains while maintaining high-quality label supervision.

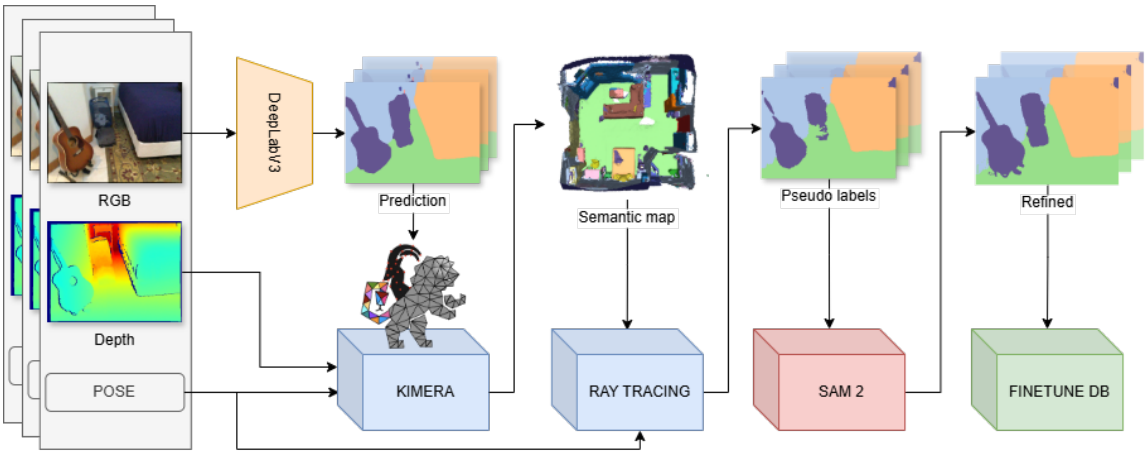


Figure 3.4: Overview of the proposed architecture. Semantic predictions and depth maps are passed to Kimera for 3D mapping, followed by ray casting. SAM refines the resulting pseudo-labels, which are then used for continual training.

3.4. Prompting Methods

SAM requires simple prompts to generate accurate high-resolution masks. We implemented and evaluated two automated prompting methods:

- **Calculated Mode (Sam erd nWm Ns 3.4.1):** For each pseudo-labeled component, meaning each connected area of pixels sharing the same label in the pseudo-label map, we compute a bounding box and its centroid. We then apply several heuristics to reduce errors caused by noisy labels and complex scenes. This method integrates semantic predictions, spatial structure, and filtering rules to refine prompts.
- **Grid Mode:** In scenes with dense or ambiguous content, we apply a regular grid of spatial points over the image. Each point serves as a prompt, and resulting SAM sub-masks are merged from largest to smallest. This method reduces dependence on pseudo-labels and provides wide coverage, especially for small or occluded objects.

3.4.1. Calculated Prompting Details

The *Calculated Prompting* method, referred to as **erd+nWm+Ns**, integrates multiple complementary strategies to enhance segmentation robustness in the presence of noisy pseudo-labels and visually complex scenes.

First, the **Fill Strategy** (*erd*, short for "ereditary") addresses cases where SAM fails to segment certain areas—often visible as black or undefined regions. In such cases, pixel values are directly inherited from the corresponding pseudo-labels. This conservative approach avoids injecting potentially erroneous guesses in low-confidence regions and provides a stable fallback.

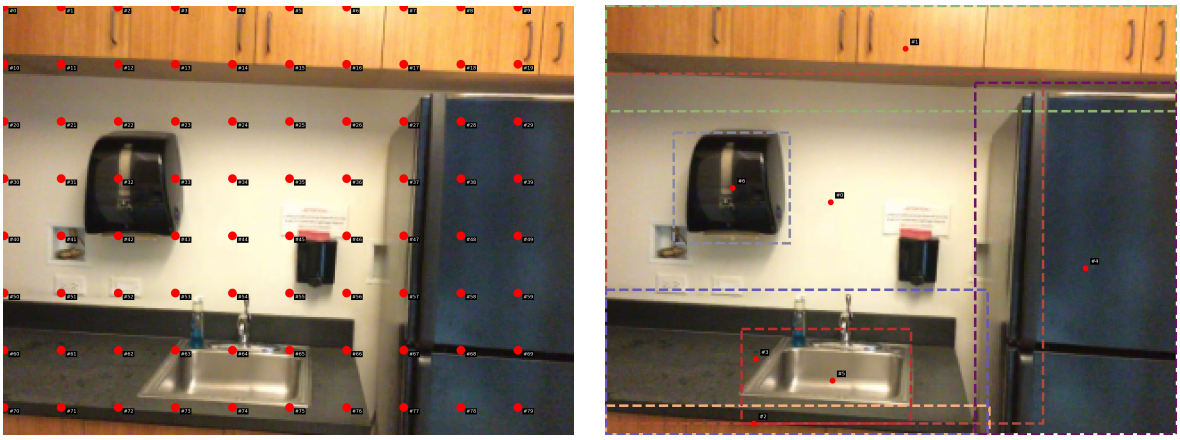
For spatial guidance, the **Centroid Calculation** is performed in a standard manner: computing the centroid of each connected component based on spatial mass within the pseudo-label. To improve alignment, the centroid is adjusted to ensure it falls within a valid region of the same label, achieving a balance between computational simplicity and precise region targeting.

The **Segmentation Region Selection** step applies a size-based filtering mechanism, denoted as *Ns* ("No small"). Components smaller than a fixed percentage of the total image area are excluded from the prompting process. This filtering helps reduce instability caused by small, scattered components, particularly in highly noisy or cluttered pseudo-labels.

In the **Label Exclusion** stage (*nWm*, "no Walls/Floor maximize"), certain semantic classes such as `wall` and `floor` are excluded from the fill strategy due to their tendency to cause over-segmentation or boundary artifacts. While initial segmentation is still performed for these categories, their labels are assigned directly rather than using a majority-label fill, since majority voting over the entire image can lead to incorrect labels caused by over-segmentation. Instead, these labels are applied first and may be overridden by other classes later in the pipeline.

Finally, the method uses a **Centroid + Box Prompting** scheme, where both the centroid and bounding box of each selected region are passed to SAM. This dual-input strategy provides the model with both central anchor points and spatial extent, improving coverage and consistency in segmented outputs.

The combination of these strategies—*erd*, *nWm*, and *Ns*—was selected after extensive empirical validation (see Section 4). It offers a strong balance between segmentation accuracy, computational efficiency, and resilience to noisy label artifacts.



Grid Prompting Calculated Prompting

Figure 3.5: Comparison of Grid Prompting (left) and Calculated Prompting (right). Grid Prompting samples inputs uniformly, while Calculated Prompting focuses on key regions.

3.5. Implementation Details

Our full pipeline is implemented using PyTorch and a ROS-based modular infrastructure, enabling real-time operation on robotic platforms. Each functional component is encapsulated as a ROS node, allowing for asynchronous, scalable deployment. A dedicated **control node** manages image flow, timing synchronization, and inter-node communication, ensuring robust real-time performance.

- **Network:** We employ DeepLabV3 with a ResNet-101 backbone, implemented in PyTorch. The model is pretrained on ScanNet [4], excluding scenes 0 to 9, and fine-tuned for adaptation assessment. The checkpoint is based on the configuration used in the paper *Unsupervised Continual Semantic Adaptation through Neural Rendering* [8].
- **Input Resolutions:** All images in the main pipeline are resized to 320×240 for efficiency. For the SAM-based segmentation refinement, RGB images are tested at both 320×240 and the original 1296×968 resolution to evaluate the impact of resolution on mask quality.
- **Voxel Sizes:** We evaluate both 3 cm and 5 cm voxel sizes. The finer resolution offers more detailed semantics, while the coarser grid reduces memory usage and computational load.

- **Mapping Engine:** The mapping is handled using Kimera [5], which internally utilizes Voxblox [9] for volumetric integration. It consumes depth images, semantic masks, and estimated camera poses to construct a 3D semantic map in real time.

- **Segmentation Refinement:** The Segment Anything Model (SAM) is integrated via its public Python API. Prompt generation is fully automated and based on geometric cues extracted from depth and spatial priors.

- **Training Loop:** We use an online continual learning strategy where the model is updated frame-by-frame using the refined semantic masks. A lightweight training module performs updates every N frames to allow fast adaptation without disrupting inference.

- **Testing Framework:** For benchmarking and ablation studies, we employ a testing pipeline that decouples runtime constraints by saving all inputs (images, poses, and depth maps) to disk. A mocked version of the runtime modules is used to replay this data, enabling consistent evaluation of timing and accuracy.

- **Evaluation Environments:** All experiments are conducted on ScanNet scenes 0–9 [4]. The pipeline is also compatible with Habitat [14], although it has not been tested in that environment yet.

- **Dataset – ScanNet:** We use the ScanNet dataset [4] as the primary benchmark for evaluation. ScanNet is a large-scale RGB-D dataset containing over 1500 indoor scenes with dense surface reconstructions and pixel-wise semantic annotations. It is widely used in 3D scene understanding research due to its diversity of room layouts, lighting conditions, and object types. Its dense semantic labels and synchronized RGB-D data make it particularly suitable for training and evaluating semantic mapping pipelines. We use scenes 0–9 exclusively for testing, as they were excluded during pretraining to ensure a clear separation between training and evaluation data and to assess generalization to previously unseen environments.

4. Experiments

4.1. Experimental setup

This chapter presents the experimental evaluation of the proposed SAM2 [10] refinement technique within a mocked pipeline developed in ROS. The experiments aim to assess the performance and adaptability of the refinement method across different voxel resolutions and image scales. To facilitate this, semantic labels were precomputed using DeepLabV3, and the mapping component was initially validated by reproducing the results of *Unsupervised Continual Semantic Adaptation through Neural Rendering* [8]. While the results are not identical, they are consistent with the original findings, thereby supporting the integrity of the pipeline. Pseudo labels were generated for voxel maps at 3 cm and 5 cm resolutions. Two main experimental campaigns were conducted: the first to explore and define a calculated prompt strategy, and the second to evaluate the effectiveness of refinement across various configurations, including voxel size (3 cm and 5 cm), image scales (large and small RGB images), and prompting strategies (calculated and grid-based). Specifications and experiment details will be discussed in the following sections.

4.2. Mapping Reproduction and Pseudo Label Generation

Even though a direct comparison with the original paper is not the main goal of this work, we leveraged the public checkpoint provided by the authors of *Unsupervised Continual Semantic Adaptation through Neural Rendering* [8], which was trained on the ScanNet25k dataset. To replicate their semantic input, we generated DeepLabV3 predictions by loading their checkpoint and running inference on RGB images from scenes 0 to 5—scenes that were not part of the model's pretraining set. For each scene, semantic results were saved in separate folders. The generated segmentations matched those in the original work, providing a solid foundation for subsequent steps.

Scenes 0–5 were specifically selected because they were excluded from the model's training set, ensuring that the generated labels are not biased by overfitting and allowing for a more objective evaluation of the pipeline. For the initial pseudo-label experiments, we focused on these six scenes to keep computational requirements manageable during development and tuning.

To proceed with the mapping phase and pseudo-label generation, we employed the `Kimera Interfacer` library by Jonas Frey. However, the original implementation required significant modifications to meet our performance and adaptability needs. We rewrote and extended the library to enhance its compatibility with our pipeline and to boost efficiency.

Initially, we generated 3D meshes using a 5 cm voxel size. While the resulting maps were coherent with those from the original work, they exhibited discrepancies that suggested a potential mismatch in voxel size specifications. After adjusting the resolution to 3 cm, we observed improved alignment with the original results, although exact reproduction remained elusive. These preliminary tests were conducted on scenes 0 through 5. A comparative summary of our mapping results and the original paper's outputs is shown in Table 4.1.

Scene	Ours			Paper	
	Deeplab	Pseudo 5 cm	Pseudo 3 cm	Deeplab	Pseudo 3D
0	41.2	48.1	48.4	41.1	48.9
1	36.0	28.8	33.0	35.5	33.9
2	23.7	26.0	27.6	23.5	25.1
3	62.9	63.9	66.8	62.8	65.3
4	49.8	42.6	49.7	49.8	49.3
5	48.7	49.3	52.2	48.9	51.7
Avg	43.7	43.1	46.3	43.6	45.7

Table 4.1: Comparison of mIoU results between our method and the original paper on scenes 0–5. Improvements are marked in green, regressions in red.

For the ray tracing component, we completely overhauled the relevant sections of the `Kimera Interfacer`, creating a new dedicated ROS node and Python library optimized for performance. Using this system, we generated semantic voxel maps by feeding in the predicted depth maps, DeepLab segmentations, and camera poses. Pseudo labels were produced by passing the constructed 3D maps and corresponding poses into the ray tracing module.

The full pipeline was executed using both 5 cm and 3 cm voxel resolutions. All results were evaluated using the mIoU metrics library provided by the authors of the original work. Already at this stage, the benefits of finer voxel resolutions (3 cm) were noticeable in terms of segmentation quality, albeit with significantly higher computational demands—both in map construction and during ray tracing.

Although we did not have access to the original pseudo labels from the authors, the trends in our results—both improvements and degradations across resolution changes—are consistent with those reported in their paper. This alignment further supports the validity of our reproduction effort, despite minor implementation-level differences.

4.3. Calculated Strategy Initial Experiments

This section presents the first series of experiments focused on evaluating the calculated prompting strategy. These experiments aimed to test and refine methods to overcome several limitations observed in the Segment Anything Model (SAM) when applied to pseudo-label refinement. Initial evaluations were conducted on scenes 0 to 5, sampling approximately one frame every 30 in each sequence. Performance was assessed using mIoU against the original pseudo labels.

4.3.1. Problems in SAM-Based Segmentation

During early testing, multiple critical issues were identified in SAM's behavior when applied to this task. These problems served as motivation for designing a diverse set of strategies to improve prompt quality and result accuracy:

- **Incomplete Masks:** SAM often fails to produce masks that fully cover the relevant areas of the image, leaving large regions unsegmented, particularly near object boundaries or in low-contrast zones.
- **Difficult Labels:** Semantic categories such as *walls* and *floors* proved particularly hard for SAM to identify accurately, likely due to their diffuse texture and large-scale presence in indoor scenes.
- **Centroid Drift:** The centroid used for prompting SAM can fall outside the intended object—especially in irregular shapes—causing incorrect label assignments (e.g., capturing a different nearby object).
- **"Picasso" Images:** A phenomenon where DeepLab or pseudo labels produce masks with excessive fragmentation—many small disconnected regions—usually due to noise or uncertain semantic boundaries. These "Picasso" cases increase computational cost and error likelihood in mask generation. Filtering out small regions can reduce the overhead and avoid reinforcing noisy labels.
- **Prompt Type Sensitivity:** While SAM can accept both bounding boxes and points as prompts, using only points was found to lead to overly narrow

segmentation focused on object fragments. Using bounding boxes alone (Obb) also did not improve performance. The standard combination of centroid and bounding box was retained as baseline.

- **Fill Strategy Errors:** When SAM produces masks that incorrectly cover large parts of the image (especially with wall labels), calculating the majority label over the entire mask can lead to mislabeling. To counteract this, some strategies override the fill label to match the original prompt category, particularly for classes known to confuse SAM.

To address these issues, multiple ablation strategies were developed and tested. These are grouped and explained below by category.

4.3.2. Unlabeled Region Fill Strategies

A common problem with SAM-based segmentation is that it does not always generate masks that cover the full extent of the object or scene. As a result, certain areas—especially object boundaries or low-contrast regions—remain unsegmented, appearing as black regions in the mask. How these unlabeled regions are handled has a significant impact on the quality of the final pseudo labels.

Two strategies were evaluated to fill these non-segmented regions:

- **Erd (Hereditary):** This approach inherits the original pseudo labels by directly copying the label values from the initial map into the areas that SAM fails to segment. It preserves prior knowledge in regions where SAM offers no prediction, reducing the likelihood of introducing new errors.
- **Max:** This strategy treats the unsegmented regions as connected components and assigns them the majority label found in their surroundings. However, this method proved highly unreliable. Since SAM often fails to segment the edges of objects, this approach frequently mislabels boundary regions and introduces significant noise into the refinement process.

Due to the consistently poor results observed with the **Max** strategy, all subsequent experiments—including variations in centroid calculation, region selection, label exclusion, and prompt configuration—used the **Erd (hereditary)** strategy as the default. This ensured that the handling of unlabeled areas remained stable and that the impact of other strategies could be assessed in isolation.

4.3.3. Centroid Calculation Strategies

- **Normal:** Calculates the centroid of the target region and moves it to the nearest pixel with a matching label in the pseudo label.
- **Ib (Inner Box):** Computes the largest bounding rectangle that fits entirely within the pseudo-labeled region. This method is more reliable but has a high computational cost and did not significantly improve results.

4.3.4. Region Selection for Segmentation

- **Normal:** Selects connected components and removes small noise via a dilated mask. Effective in general, but computationally heavy for "Picasso" images.
- **Ns (No Small):** Filters out components below a pixel threshold to eliminate noise and reduce mask complexity. While rough, it avoids the over-segmentation typical of noisy pseudo labels.
- **Nm (No Multiple):** Rejects regions covering multiple labels. Although promising in theory, this strategy fails when large correct segments happen to include small irrelevant components, leading to undersegmentation.

4.3.5. Label Exclusion Strategies

- **None:** Includes all labels—even difficult ones like walls and floors. This often improves edge alignment but increases the risk of mislabeling from classes SAM struggles with.

- **nW-nWF:** Excludes problematic classes from prompting and evaluation. This approach reduces false positives but also weakens the completeness of segmentation and did not yield worthwhile improvements.

4.3.6. SAM Prompt Configuration

- **Normal:** Uses both bounding box and centroid as prompt input to SAM.
- **Obb (Only BBox):** Explores the use of bounding boxes alone. Intended to encourage holistic segmentation, but yielded no benefits.
- **Op (Only Points):** Rejected based on prior experience showing it leads to focus on small image fragments rather than full object regions.

4.3.7. Fill Strategies

- **Normal:** Assigns the majority label in the mask region.
- **nWm - nWFm:** For known problematic classes (e.g., walls), overrides majority fill and instead forces the label used to generate the prompt, reducing mislabeling when SAM's mask spans the whole image.

Due to the poor performance of the **Max** strategy—where the majority label was incorrectly assigned to unsegmented regions—all further tests adopted the **Erd (hereditary)** strategy as the default for handling unsegmented pixels. This choice allowed the evaluation to focus on the impact of other components in the pipeline (centroid calculation, region selection, label filtering, etc.), without noise from faulty fill methods.

All tests were conducted using the 3 cm pseudo labels as reference and one frame every 30 in each scene. Table 4.2 shows the mIoU results for various combinations built on the Erd baseline. While differences are often subtle, strategies involving region selection and label filtering showed potential to mitigate SAM's failure cases. If not specified otherwise, the normal strategies were used as the default.

Scene	Pseudo 3 cm	SAM Max	SAM Erd	SAM Erd nW	SAM Erd lb	SAM Erd Ns	SAM Erd nWm
0	48.4	44.9	48.4	48.3	49.0	48.9	48.4
1	33.0	32.6	33.2	33.3	33.2	32.9	33.2
2	27.6	28.1	29.0	28.9	28.5	29.0	29.3
3	66.8	61.7	62.3	62.4	62.3	62.3	62.6
4	49.7	48.9	50.8	48.7	50.7	50.9	50.7
5	52.2	50.7	52.3	51.9	52.7	52.3	52.2

Table 4.2: Evaluation of refinement strategies using Erd as base. Pseudo 3 cm serves as reference. Improvements over the reference are highlighted in green, regressions in red.

Among these, the strategy that consistently performed well both numerically and logically was the combination of **Erd + Ns + nWm**. This configuration jointly addresses unsegmented regions, label noise in fragmented pseudo maps, and risky labels that tend to dominate incorrect segmentations. The full evaluation results for this optimal configuration are shown in Table 4.3.

Scene	Pseudo 3 cm	SAM Erd Ns nWm
0	48.4	48.6
1	33.0	33.5
2	27.6	27.7
3	66.8	66.0
4	49.7	51.2
5	52.2	53.6
Avg	**46.3**	46.8

Table 4.3: Final test results using Erd + Ns + nWm configuration across scenes 0–5. Improvements over pseudo labels are marked in green, regressions in red.

It is important to note that the results discussed in this section were obtained by sampling approximately one out of every 30 frames in each scene. While useful for rapidly testing and comparing different strategies, this sparse sampling made the evaluation sensitive to outlier images—such as those affected by severe pseudo label noise or segmentation failures (e.g., "Picasso" images). As a result, some strategies may have appeared less effective than they truly are. In the following section, where the full dataset is used for evaluation, the refinement methods—particularly those based on calculated prompting—show significantly more robust, and consistent improvements.

4.4. Evaluation and Experimental Protocol

To ensure consistency and comparability, the evaluation methodology closely follows the setup used in *Unsupervised Continual Semantic Adaptation through Neural Rendering* [8]. The primary metric used is mean Intersection-over-Union (mIoU), which quantifies the overlap between predicted labels and ground truth on a per-class basis, averaged across all valid classes.

The performance is measured by computing the mIoU between the refined pseudo labels and the ScanNet ground truth. To isolate the effect of the refinement pipeline, the results are always reported relative to the original (unrefined) pseudo labels. This approach highlights whether the refinement step contributes with a meaningful improvement to the segmentation quality.

The experiments are conducted on approximately 80% of the valid RGB-D frames in each scene. A frame is considered valid if the corresponding ScanNet ground truth labels are non-empty and usable. Some scenes in the ScanNet dataset contain invalid frames, where ground truth annotations are missing or entirely black due to occlusions or sensor tracking errors. These frames are excluded from evaluation.

All evaluations are performed using the exact same codebase and mIoU computation script provided by the authors of the original paper [8], ensuring that our results are directly comparable. It is important to note that ScanNet's ground truth is itself derived from reconstructed 3D meshes and may contain inaccuracies. As a result, our reported mIoU scores may underestimate true performance, and the relative improvements from refinement could be even more significant than they appear.

4.5. General Experimental Evaluation

To comprehensively evaluate the performance of the SAM refinement pipeline, we conducted tests across multiple configurations. Specifically, we compared the **calculated** and **grid** prompting strategies on both 3 cm and 5 cm voxel resolutions,

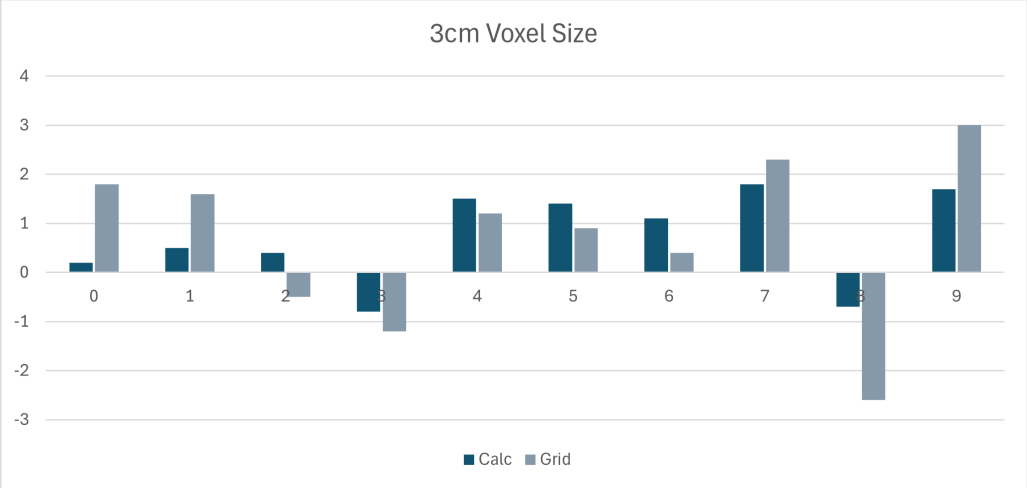
using two RGB input image sizes: high-resolution (1296×968) and low-resolution (320×240). The experiments were performed on scenes 0 to 9, using 80% of valid frames.

4.5.1. Quantitative Results

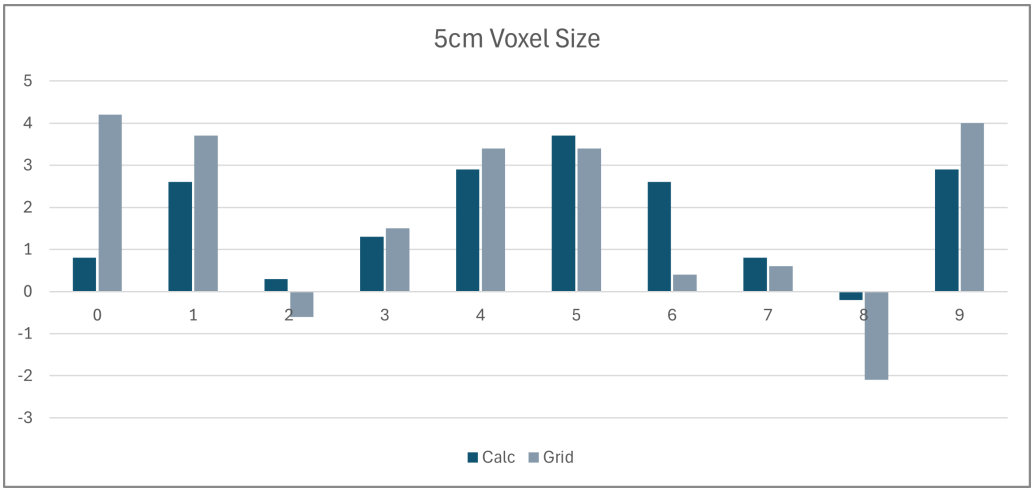
Figure 4.1 shows per-scene improvements in pseudo-label quality (measured via mIoU) at high resolution, comparing the *calculated prompting* (Calc) and *grid prompting* (Grid) strategies across 3 cm and 5 cm voxel sizes. Each bar represents the refinement effect of SAM on a specific scene, revealing variability across scenes, prompting strategies, and resolutions.

Overall, SAM refinement consistently improves pseudo-label quality. As expected, the 5 cm voxel configuration yields greater improvements than 3 cm. Coarser voxel grids tend to tolerate minor segmentation errors better and benefit more from smoothing, which aligns with the higher average improvements observed.

Grid prompting often results in slightly higher average improvements but also introduces greater variance across scenes. Notably, Scene 7 exhibits a clear performance drop under grid prompting in both resolutions—an issue explored in detail in the next subsection on qualitative results. This suggests that grid prompting, which engages more pixels and relies more heavily on SAM's segmentation predictions, is more vulnerable to failure in specific scene types, particularly large, uniform, or texture-poor regions.



Difference in mIoU from pseudo-labels – 3 cm Voxel Resolution



Difference in mIoU from pseudo-labels – 5 cm Voxel Resolution

Figure 4.1: Per-scene improvements using the `Diff Big` metric with calculated and grid prompting strategies. The 5 cm grid shows greater overall improvement but less consistency across scenes, especially in Scene 7.

Table 4.4 provides a summary of mean and standard deviation values for each prompting method and voxel resolution. While grid prompting achieves the highest average improvement at 5 cm (1.85 vs. 1.77), it also shows the highest variability (std. dev. 2.09 vs. 1.26). This supports earlier observations of scene-specific instability.

Calculated prompting shows more consistent results, particularly with high-resolution (1296×968) RGB inputs. Compared to lower-resolution inputs (320×240), the standard deviation of calculated prompting drops from 1.27 to 0.90 at 3 cm, and from 1.32 to 1.26 at 5 cm. In contrast, the grid strategy exhibits a slight increase in variance with higher input resolution—about 0.1—indicating negligible impact.

Taken together, these results support using **calculated prompting at 5 cm voxel resolution** as a performance and efficient baseline. It achieves competitive improvement while maintaining lower variability across scenes, offering a stable middle ground between performance and reliability. While high-resolution inputs improve consistency—especially for the calculated strategy—the absolute gains over low-resolution inputs are modest. Given that low-resolution inputs require significantly less computational overhead and faster processing, they remain a practical alternative in resource-constrained settings where runtime efficiency is a priority.

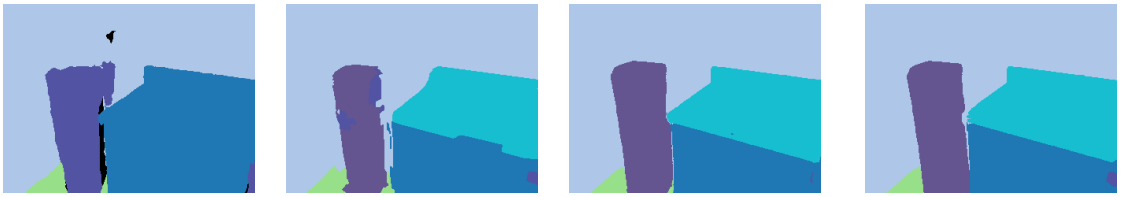
Voxel Size	Prompting Strategy	Mean ΔmIoU	Std. Dev.
3 cm	Calculated (Calc)	0.71	0.90
	Grid (Grid)	0.69	1.62
5 cm	Calculated (Calc)	1.77	1.26
	Grid (Grid)	1.85	2.09

Table 4.4: Mean and standard deviation of mIoU improvement across all scenes, comparing Calculated and Grid prompting strategies at two voxel sizes.

4.5.2. Qualitative Results

To complement the quantitative findings, we present visual comparisons of segmentation results across different refinement strategies. All examples in this subsection use the configuration that yielded the best performance in prior experiments: **5 cm voxel resolution** with **high-resolution RGB input** (1296 × 968).

These comparisons highlight improvements in boundary accuracy, semantic consistency, and noise reduction. Each example includes four images: the ScanNet ground truth, the original pseudo labels, and the refined outputs using SAM with both grid and calculated prompting.



Ground truth Pseudo label Grid prompting Calculated prompting

Figure 4.2: Successful refinement example. Both strategies improve label consistency and object boundaries.

In many cases, one prompting strategy performs better than the other. Below are examples where calculated prompting outperforms grid prompting and vice versa.

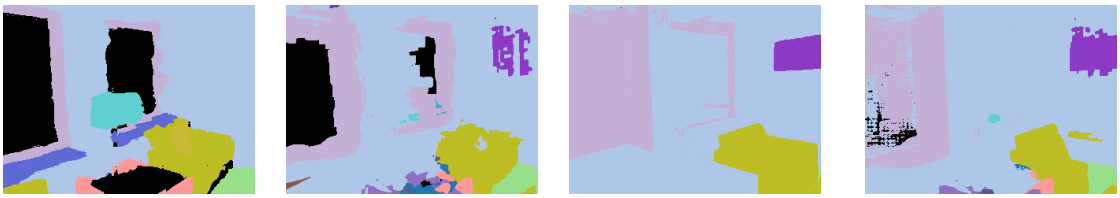
Ground truth Pseudo label Grid prompting Calculated prompting

Figure 4.3: Calculated prompting performs better: grid-based segmentation overextends into incorrect areas, mislabeling large regions.

Ground truth Pseudo label Grid prompting Calculated prompting

Figure 4.4: Grid prompting performs better: a noisy pseudo label leads the calculated prompt to erase the door region entirely.

In rare but important cases, both refinement methods fail due to incorrect mask predictions from SAM itself. These situations result in the deletion of entire objects or propagation of noise.



Ground truth | Pseudo label | Grid prompting | Calculated prompting

Figure 4.5: Failure case: SAM fails to identify key objects, and both refinement methods propagate incorrect labels that erase valid content.

These examples highlight the trade-offs between the two prompting strategies. Grid prompting tends to be more aggressive, achieving stronger improvements in some cases but also being more vulnerable to failure. Calculated prompting is more conservative and stable, but its accuracy relies heavily on the initial pseudo label quality. Selecting the appropriate strategy depends on the application's tolerance for risk and computational constraints.

4.6. Discussion

The experimental results presented in this chapter demonstrate the effectiveness and complexity of using the Segment Anything Model (SAM) for refining 3D pseudo labels through image-space prompting. Across all configurations, SAM-based refinement consistently improved the quality of pseudo labels, with varying performance depending on voxel size, image resolution, and prompting strategy.

One of the most striking findings is that the improvement from refinement was more noticeable on 5 cm voxel maps. Although these maps are coarser and capture less geometric detail than 3 cm maps, they tend to introduce more initial noise due to voxel quantization. As a result, SAM refinement is able to correct larger errors, leading to more significant relative gains in mIoU. This makes the 5 cm configuration especially attractive for real-time applications, offering both improved accuracy and reduced computational cost compared to higher-resolution voxel maps.

The choice of prompting strategy played a major role in determining refinement quality. Calculated prompting produced stable and conservative improvements by leveraging prior pseudo labels and region-specific cues. It was particularly effective

when the pseudo labels were spatially coherent. Grid prompting, on the other hand, performed more aggressive corrections by densely sampling the image, often leading to larger improvements—but at the cost of stability. Its standard deviation across scenes was roughly twice that of the calculated strategy, indicating more frequent failures when SAM masks were incorrect.

The input image scale also had a measurable impact. Using high-resolution RGB images (1296×968) consistently improved refinement accuracy over low-resolution images, especially in finer voxel maps. However, the gains were modest, and the increase in computational time makes this a decision that must be balanced based on system constraints.

Failure cases occurred in both strategies, often triggered by poor pseudo label structure ("Picasso" scenes) or semantic ambiguity. SAM occasionally failed to segment critical objects—such as doors or chairs—or oversegmented background classes like walls. These failures could cascade through the refinement pipeline, resulting in the deletion of important labels or the propagation of incorrect regions. The ablation study of fill strategies further confirmed that improper handling of unsegmented areas, such as with the Max fill method, could exacerbate these issues. Consequently, the Erd (hereditary) strategy was adopted as the baseline to avoid unnecessary corruption of label information.

An unexpected insight was how sensitive SAM's output can be to prompt placement. Slight misalignments in centroid selection or excessive overlap with multiple labels could lead to total segmentation failure. This reinforces the importance of careful prompt calculation and filtering in any practical deployment.

Overall, the SAM refinement pipeline is a powerful but sensitive tool. Its effectiveness depends heavily on how well its components—prompts, voxel resolution, fill strategy, and image input—are tuned to the scene and dataset. The system shows clear potential for improving label quality in noisy or incomplete maps, and particularly excels in conditions where ground truth annotations are weak or unavailable.

Future improvements could include incorporating uncertainty estimation into prompt selection, combining SAM with class-aware refinement constraints, or introducing feedback mechanisms to reject failed segmentations. Dynamic prompting in online systems could further enhance its utility in real-time robotics or SLAM pipelines.

5. Conclusion

In this thesis, we presented a novel approach to refining pseudo-labels in Unsupervised Continual Domain Adaptation (UCDA) pipelines by integrating the Segment Anything Model (SAM). Building upon the existing 2D–3D pseudo-label fusion framework using Kimera and DeepLabV3, we proposed a SAM-based refinement module that improves the quality of projected 3D pseudo-labels through targeted 2D segmentation.

Two prompting strategies were explored: a conservative calculated prompting method based on centroid and label heuristics, and a more aggressive grid prompting technique. Extensive experimentation across voxel resolutions, image scales, and prompt configurations demonstrated that SAM refinement yields measurable improvements in pseudo-label accuracy, particularly for noisy or low-resolution semantic maps.

A key insight was that refinement is especially effective when applied to 5 cm voxelmaps, where coarser geometry introduces more initial label noise. While 3 cm maps offer higher fidelity, the added benefit of SAM refinement is less pronounced. The conservative prompting strategy achieved more stable results, while grid prompting offered higher peak performance at the cost of consistency. These trade-offs were clearly observed in both quantitative metrics and qualitative failure cases.

We also contributed a reimplementation and performance-optimized version of the Kimera interfacer library, improving its usability and integration with modern ROS pipelines. Our ablation studies further explored how fill strategies, region selection, and label exclusion influence the final outcome, confirming that robust handling of unsegmented areas is critical to prevent error propagation.

Future work will extend this approach by integrating SAM-based refinement into a fully online continual training loop. Preliminary experiments combining refinement with continual fine-tuning indicate promising gains in model adaptability. We also aim to explore uncertainty-based prompt filtering and the incorporation of class-aware segmentation priors to make refinement more robust and context-sensitive.

Ultimately, this thesis demonstrates that large-scale segmentation models like SAM can play a valuable role in enhancing weak supervision signals in UCDA settings. With further optimization and integration, this technique holds strong potential for improving real-world robotic perception and continual learning systems.

Bibliography

[1] Nikita Araslanov and Stefan Roth. Self-supervised augmentation consistency for adapting semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[2] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes, 2017.

[5] Jonas Frey, Hermann Blum, Francesco Milano, Roland Siegwart, and Cesar Cadena. Continual adaptation of semantic segmentation using complementary 2d-3d data representations. *IEEE Robotics and Automation Letters*, 7(4):11665–11672, 2022.

[6] Joonhyuk Kim, Sahng-Min Yoo, Gyeong-Moon Park, and Jong-Hwan Kim. Continual unsupervised domain adaptation for semantic segmentation, 2021.

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Laura Gustafson Rolland, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[8] Zhizheng Liu, Francesco Milano, Jonas Frey, Roland Siegwart, Hermann Blum, and Cesar Cadena. Unsupervised continual semantic adaptation through neural rendering, 2023.

[9] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 1366–1373. IEEE, September 2017.

[10] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024.

[11] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[12] Alexandru Stan and Mohammad Rostami. Source-free continual semantic segmentation via latent distribution alignment. In *CVPR*, 2024.

[13] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[14] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, Alexander William Clegg, and Devendra Singh Chaplot. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4927–4936, June 2023.

[15] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.