Configurable Embodied Data Generation for Class-Agnostic RGB-D Video Segmentation

Anthony Opipari[1], Aravindhan K Krishnan[2], Shreekant Gayaka[2], Min Sun[2]
Cheng-Hao Kuo[2], Arnie Sen[2], Odest Chadwicke Jenkins[1]

Fig. 1: Illustration of the data generation process used to create RGB-D videos and associated segmentation labels in the MVPd dataset. Left: Videos are collected from 3D reconstructions [1] of real-world building-scale domestic environments containing clutter, heavy occlusion, and diverse objects. Right: The videos are configured to represent specific robot embodiments based on sensor placement (e.g. 'tall' or 'short' form factors as shown by pink and blue trajectories), sensor type, and illumination source (e.g. active or ambient).

***Abstract*— This paper presents a method for generating large-scale datasets to improve class-agnostic video segmentation across robots with different form factors. Specifically, we consider the question of whether video segmentation models trained on generic segmentation data could be more effective for particular robot platforms *if* robot embodiment is factored into the data generation process. To answer this question, a pipeline is formulated for using 3D reconstructions (e.g. from HM3DSem [1]) to generate segmented videos that are configurable based on a robot's embodiment (e.g. sensor type, sensor placement, and illumination source). A resulting massive RGB-D video panoptic segmentation dataset (MVPd) is introduced for extensive benchmarking with foundation and video segmentation models, as well as to support embodiment-focused research in video segmentation. Our experimental findings demonstrate that using MVPd for finetuning can lead to performance improvements when transferring foundation models to certain robot embodiments, such as specific camera placements. These experiments also show that using 3D modalities (depth images and camera pose) can lead to improvements in video segmentation accuracy and consistency.**
Project Page: https://topipari.com/projects/MVPd

I. Introduction

Semantic mapping remains a critical but daunting challenge for developing robots that can understand and function autonomously in open-world human environments. A part of this challenge stems from object-level mapping and specifically the need for robots to perceive the form and function of a vast distribution of objects that appear in domestic settings [2]. Further confounding the problem are environmental factors such as occlusions, obscure lighting, affordances, and co-occurrences that increase the set of circumstances robots encounter. Such environmental diversity is a staggering obstacle to the utility of applying data-driven object segmentation methods to mapping. This has led to *increasingly large datasets* and foundation models tailored to image segmentation [3], [4]. On the other hand, robots and their environments are both dynamic—their perspective and the position of objects in view frequently change, which motivates use of *video segmentation* in-place of static image segmentation methods. Finally, the space of *robot embodiments* is itself another factor limiting scene-understanding since training methods with data from one robot could suffer from decreased performance when transferred to a different robot with a distinct embodiment. These observations lead to the central question addressed in this paper: **'How can roboticists scalably collect video segmentation datasets that are configurable based on their robot's specific embodiment without expensive field collection?'**

Longstanding image segmentation tasks from computer and robot vision [5], [6] have been extended to the video domain in which the goal is to accurately predict the set of pixels belonging to certain objects within each frame of a video [7]–[9]. In part, this is driven by the need for *temporally consistent* as well as accurate segment predictions in many applications including those in robotics [10]. This paper focuses on *class-agnostic* video instance segmentation, in which algorithms are expected to predict segments for *every instance of every object* and maintain temporal consistency of the predictions throughout a video. Class-agnostic video segmentation in the open-world is especially critical for domestic robots that operate in cluttered home environments containing a wide distribution of object categories, not all of which can be included in finite training datasets. A natural consequence of moving to the video domain is an associated increase in the expense of collecting videos

[1]University of Michigan, {topipari, ocj}@umich.edu
[2]Amazon Inc., {krsar, sgayaka, minnsun, chkuo, senarnie}@amazon.com

with densely annotated segmentation masks [8], [11]. Thus, in contrast to the image domain, for which many large-scale datasets have been published [3], [12], [13], there are relatively few comparably sized video segmentation datasets.

This work sets out to provide a scalable solution for creating video segmentation datasets that are customised to specific robot embodiments. A key insight inspring the work is the increasing availability of 3D reconstructions [1], [14], [15] of real-world, cluttered, domestic environments and *the potential for these representations to be used to control for robot embodiment while generating large-scale and densely annotated segmentation videos*. With this insight, the present paper makes the following contributions:

1) The **Massive Video Panoptic dataset (MVPd)** and data generation pipeline for configurable robot video segmentation data. In total, MVPd is *more than 45× larger* than existing video segmentation benchmarks with **18K annotated RGB-D videos, >6M images, and 162M masks**. The data generation pipeline can create large-scale simulated datasets controlling for embodiment such as sensor placement and scene illumination.
2) Benchmarking experiments are conducted to evaluate state-of-the-art algorithm performance using MVPd on the class-agnostic video instance segmentation task. Results are included to compare both foundation models and specially designed video segmentation models.
3) Extensive ablation experiments are carried out to establish the impact of sensor placement on segmentation quality as well as the potential for 3D data, in the form of depth images and camera pose, to improve segmentation consistency. Presented findings show that **configuring for embodiment during the training data generation process can improve segmentation quality and consistency across distinct robot embodiments**.

II. Related Work

Video segmentation can be summarized into two tracks: a semantic track and a class-agnostic track. Within the semantic track, video semantic segmentation expects pixel-level segments for each object in each video frame along with a label for their semantic categories while video instance segmentation distinguishes between multiple objects of the same category [8]. Video panoptic segmentation considers cases in which certain categories are nebulous in shape such as the sky, floor, or 'stuff' [9]. Most methods proposed for the semantic track specialize to specific semantic categories [11], [16]–[20]. For instance, Video K-Net [17] learns a kernel for each class and differentiates among the kernels of distinct instances. Similarly, transformer-based architectures such as PAOT [19] and Tube-Link [20] have been proposed to associate features across images based on which category each feature belongs to. When algorithms are expected to segment objects *regardless of their semantic class*, *class-agnostic* video segmentation is considered [21]. A special case is video object segmentation [7], in which only one or a few specific objects of interest, chosen by their foreground position [22] or manual masking [23], [24], are to be segmented. Within class-agnostic video segmentation, much of the work uses motion cues like optical flow to separate each object [21], [25]. Many approaches consider only video object segmentation, where foreground motion is feature-rich, and hence do not segment *every* object in the scene. For example, SegGPT [26] segments a specific object of interest using an image-level foundation model and in-context coloring. For segmenting all objects in the autonomous vehicle setting, Siam et al. [21] propose using object motion estimation. Inspired by the work to use optical flow as a feature, the present paper considers whether a robot's egocentric motion (i.e. pose estimated by odometry), can be used as a feature within class-agnostic segmentation.

Open-world segmentation has been a growing topic of interest for image-level segmentation [27]–[31]. A few recent works have carried the topic into video segmentation [19], [21], [25], [32]–[34]. Again, optical flow and object motion estimation have been used as features to distinguish foreground and background objects in zero-shot applications [21], [25], [32]. More recently, Xu et al. [19] demonstrate the challenge for generalizing video segmentation to unseen categories in videos collected from YouTube. Wang et al. [35], [36] and Li et al. [37] propose using language features to extend video segmentation models to the open-world. In contrast, we set out to explore the potential for open-world video segmentation in domestic environments by embodied robots. We are inspired by recent foundation models tailored to zero-shot transfer [3], [38]. In particular, promptable models such as Segment Anything Model (SAM) [3] and FastSAM [4] have led to a number of follow-on works demonstrating their potential as transferrable models. Cen et al. [39] showed how recursive prompting to SAM with neural radiance fields can be used to extract volumetric segments from static scenes. We take inspiration from this work and consider how promptable segmentation can be used recursively *over time* in class-agnostic video segmentation.

Large-scale datasets for video segmentation are expensive to collect and annotate, which has been identified as a limiting factor for learning-based solutions [8], [11]. Despite these challenges, a number of benchmark datasets have been introduced [9], [11], [19], [40]–[43] and are summarized in Tab. I. For exterior road settings, Weber et al. [40] introduced KITTI-STEP and MOTChallenge-STEP, totaling >21K annotated images. Beyond autonomous driving, datasets collected 'in the wild' have led to even larger video segmentation datasets. Yang et al. [8] used videos from YouTube for video instance and object segmentation tasks. More recently, Miao et al. [45] introduced the VSPW dataset for video semantic segmentation before adding additional labels to support video panoptic segmentation in VIPSeg [11] and zero-shot objects by Xu et al. [19]. For class-agnostic video segmentation, Wang et al. [43] introduced the $\text{UVO}_{\mathcal{D}}$ dataset which focuses its annotations on kinetic human-object interactions. In contrast to the existing video segmentation benchmarks, MVPd provides 3D input modalities in the form of ground truth depth images and camera pose for every RGB image. In addition, MVPd focuses on indoor

Dataset	Setting	Modality	Camera Pose	Videos	Images	Obj. per Video	Classes	Stuff Categories	Unseen Split	Sensor Placement Control
Cityscapes VPS [9]	Roads	RGB-D†	×	500	3,000	28.79*	19	✓	×	×
KITTI-STEP [40]	Roads	RGB	×	50	19,103	53.76*	19	✓	×	×
MOTC-STEP [40]	Pedways	RGB	×	4	2,075	38.00*	7	✓	×	×
YTVIS [41]	Wild	RGB	×	4,046	128,930	2.10*	40	×	×	×
OVIS [42]	Wild	RGB	×	901	62,641	5.90*	25	×	×	×
VIPSeg [11]	Wild	RGB	×	3,536	84,750	13.65*	124	✓	×	×
VIPOSeg [19]	Wild	RGB	×	3,149	75,022	13.65*	125	✓	✓	×
LV-VIS [35]	Wild	RGB	×	4,828	111,298	5.3	**1196**	✓	✓	×
UVO_D [43]	Kinetics	RGB	×	1,017	91,530	13.52	1	×	✓	×
MVPd	Domestic	RGB-D	✓	**18,000**	**6,055,628**	**94.39**	40	✓	✓	✓

TABLE I: Comparison of related video instance and panoptic segmentation benchmarks with MVPd. †Noisy depth can be computed from stereo images in Cityscapes dataset [44]. MVPd provides dense depth as measured by a Matterport scanner. *Denotes calculation based only on publicly available data (i.e. not including private evaluation data).

domestic environments as opposed to exterior settings [9], [40], or videos from the wild [11], [19], [41]–[43]. The substantial object clutter observed in MVPd (94.39 objects per video) suggests it better captures unstructured domestic environments than existing video segmentation benchmarks. A separate line of work has used increasingly available 3D scanners to collect and annotate 3D point and mesh-based datasets from real-world spaces [1], [14], [15], [46], for use in 3D segmentation benchmarks. Using 3D mesh datasets, Eftekhar et al. [47] introduced a pipeline to create 'steerable' datasets for specific computer vision tasks and demonstrated their benefit on non-embodied image-based tasks. Building upon these ideas, the MVPd data generation pipeline focuses on video segmentation specifically and controlling for the features of embodiment (sensor type, sensor placement and illumination sources) that can impact video segmentation consistency and accuracy for downstream domestic robots.

III. MVPd: Massive Video Panoptic Dataset

MVPd is introduced to support research on embodied class-agnostic video instance segmentation and the potential for 3D modalities to benefit video segmentation algorithms. In total, MVPd contains 18,000 densely annotated RGB-D videos, 6,055,628 individual image frames with ground truth 6DoF pose, and 162,115,039 masks. Each video in MVPd contains between 100 and 600 image frames rendered at 640x480 resolution. Videos are captured from scenes of HM3DSem [1], containing real-world building-scale domestic environments such as homes, offices, and retail spaces. Each scene is on the scale of a multi-floor building with on average >14 rooms per scene, and 60 objects per room [1]. This is the largest semantically annotated indoor scene dataset we are aware of, covering >20,000m^2 of navigable area and >2x the number of unique object instances as are available in comparable scene datasets. Annotated segments are assigned to one of 40 Matterport categories [15].

A. Data Generation Pipeline

The key insights inspiring MVPd's data generation pipeline are that (1) *instance annotations at the mesh-level enable inexpensive segment annotations at the video-level* and (2) *the mesh representation for scenes enables embodiment-specific configuration for each video at scale*. As a source of real-world scenes, the HM3DSem dataset [1] provides mesh representations of 216 real-world building-scale environments created with a Matterport scanner. Alternative scene datasets, such as Matterport3D [15], are compatible with our data generation pipeline and offer an added source of annotated scenes that could be rendered as videos alongside MVPd in the future.

As illustrated in Fig. 2, the data generation pipeline takes input meshes that contain RGB color and segmentation textures. Using the mesh, random paths of sparse waypoints are planned using a collision-free NavMesh planner from Habitat-Sim [48] which are then interpolated by the pipeline to form trajectory plans of smooth and dense wayposes. Specifically, the pipeline uses linear interpolation to ensure the trajectories are smooth to within 5cm of linear displacement and 0.5°of axis-angle rotation. Next, an embodiment configuration file is used to refine the trajectory before rendering. According to the embodiment specification, an RGB-D camera is placed at each waypoint pose to render a corresponding video frame that may include RGB, depth, and panoptic labels. The embodiment specification may control

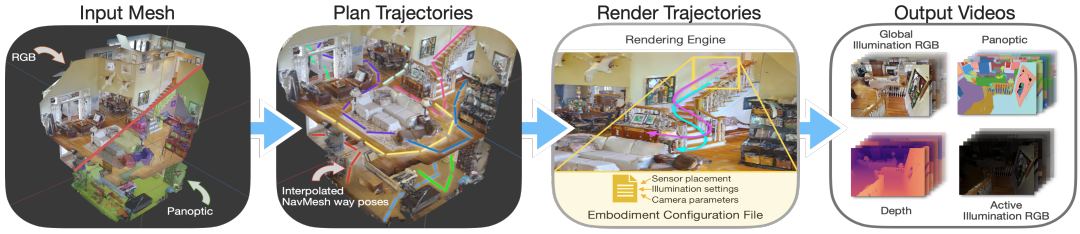

Fig. 2: Illustration of the data generation pipeline used to create MVPd. Left: Using input meshes that contain RGB and segmentation textures, the pipeline generates sparse random paths with a collision-free NavMesh planner [48] and interpolates them into dense trajectories of way poses. Right: The motion trajectories are refined according to an embodiment configuration file and rendered to output videos.

Fig. 3: Impact of specific embodiments (sensor placement & active illumination) on visual features rendered by the MVPd

for sensor placement and illumination source (i.e. ambient or active) and power (Watts) as shown in Fig. 3.

MVPd videos are rendered as follows: 180 scenes from HM3DSem are chosen for inclusion based on public availability. For each scene, 50 random start and end waypoint pairs are chosen without replacement for path planning, resulting in 50 trajectories per scene. For each trajectory, embodiment specification is set at both 1m and 0.1m above the floor to emulate challenging perspectives taken by home robots. Thus, 100 videos are rendered for each of the 180 scenes with an average trajectory distance of 7.48m.

B. Zero-Shot Subset: Seen and Unseen Categories

Zero-shot learning refers to the application of machine learning models on data categories not seen during training (the 'open-world') [27]. It is especially relevant for robots deployed in the real-world and it encompasses any learning-based task including detection, segmentation, *and* video segmentation. A zero-shot subset of MVPd is included to support research on generalizable video segmentation. Evaluations on the zero-shot subset are intended to reflect an expectation of model performance in the open-world.

The zero-shot subset of MVPd is defined by a set of '*seen*' and '*unseen*' object classes. Only the seen class are available to models during training, while both the seen and unseen classes are used for testing. To select these classes, we follow a process similar to [31], [49]. All object instances within MVPd belonging to the 'Misc' and 'Objects' Matterport categories are considered for the unseen class since these encompass a broad collection of instance morphologies unlike the remaining 38 Matterport super-categories. Next, for each object instance a CLIP-embedding (ViT-L/14) [50] is generated using the instance's corresponding human-annotated text description. The embeddings are then clustered by the k-means algorithm with 20 clusters. 20% of the clusters are assigned to the unseen set and the remaining clusters to the seen set. The split of clusters is based on the number of training videos that observe objects from the various cluster, with the least frequently observed clusters chosen for the unseen class. As illustrated in Fig. 4, the unseen class include objects relating to clothes storage (e.g. closet, shelf, cubby), hobby items (e.g. pianos, aprons, aquariums), stands (e.g. book rack, computer tower, foot stand), and soap (e.g. detergent, washing powder). Videos containing any object from the zero-shot unseen class are excluded from the training set to create a zero-shot subset.

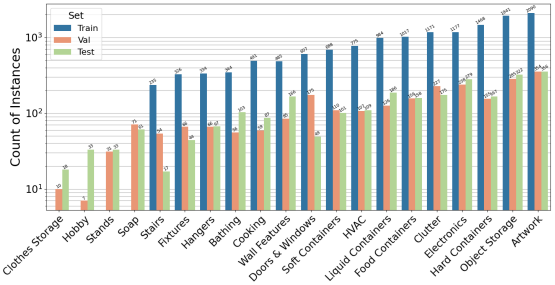

Fig. 4: Distribution of the 20 object clusters used for creating the zero-shot subset of MVPd. Object instances from the 'Objects' and 'Misc' super-categories in MVPd, as defined in [15], are grouped into 20 clusters using k-means with the CLIP-embedding [50] of each instance's human-annotated text description. Clusters are then ranked by the number of associated training videos, and the smallest 20% (clothes storage, hobby items, stands, and soap) are defined as the zero-shot classes. The text summary of each cluster (e.g. Clothes Storage, Hobby, etc.) are defined based on manual inspection.

IV. Class-Agnostic Video Instance Segmentation

Problem Definition: Given a video sequence consisting of T frames, consider a temporal window of $k \leq T$ consecutive frames denoted by $I^{t:t+k} = \{I^t, I^{t+1}, \ldots, I^{t+k}\}$. Following the definition of VPS [9], a tube prediction corresponding to the k-span window is defined as a track of frame-level segments, $\hat{u}_{z_i} = \{\hat{s}^t, \ldots, \hat{s}^{t+k}\}_{z_i}$ where z_i represents a unique instance identifier. Ground truth segment tubes are defined analogously using annotated segments at each frame in the window. The goal of class-agnostic video instance segmentation is to accurately segment every instance of every object within a video, regardless of the objects' semantic categories. Unlike VPS [9] and VIS [8], class-agnostic video instance segmentation does not require predicted tubes be assigned to one of a predefined set of semantic classes.

Evaluation Metric: A growing set of foundation segmentation models including SAM [3] and FastSAM [4] have been proposed for broad applicability and this paper set out to include them in its evaluations. However, because these models produce overlapping predictions they cannot be directly evaluated by the video panoptic quality metric introduced by Kim et al. [9] as it requires no-overlap between predicted segments [51]. To build upon the video panoptic quality metric, we propose a slight modification to enable its evaluation of models regardless of their predictions' overlap. For this modification we borrow inspiration from class-agnostic instance segmentation metrics used for images [31], [52] to define class-agnostic Video Instance Segmentation Quality (VSQ) as follows:

For a fixed k-frame window size, VSQ^k is computed by measuring the overlap between each temporally aligned ground truth and predicted segment tube of length k. As illustrated in Fig. 5, VSQ^k is computed by first matching each ground truth tube to each predicted tube using the Hungarian algorithm. The resulting assignment maximizes the sum total F-measure over each tube match. Each matched tube is considered a true positive (TP). Any predicted tube that is left unmatched is considered a false positive (FP)

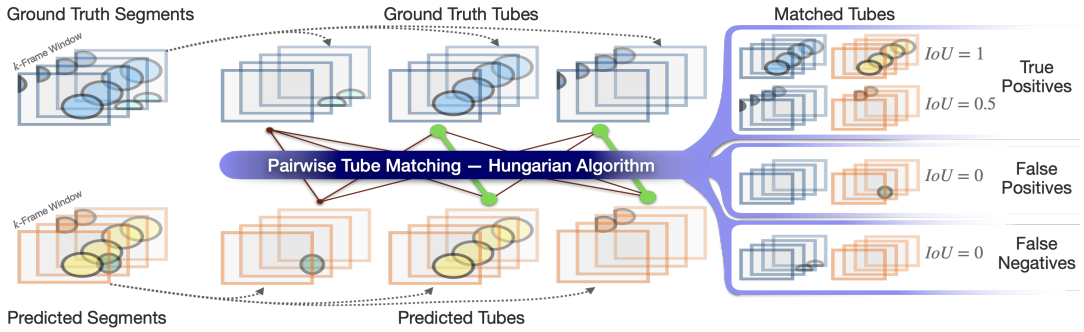

Fig. 5: Illustration of the the class-agnostic video segmentation quality (VSQ) metric. Given a fixed k-frame window size, ground truth and predicted segments are isolated into segment tubes and then matched using an optimal assignment algorithm according to pairwise F-measure. Once matched, the set of true positive, false positive and false negative tubes are counted and a per-match IoU is computed.

while any unmatched ground truth tube is considered a false negative (FN). Using the pairwise intersection over union (IoU) between each TP, VSQ^k is calculated as follows:

$$\text{VSQ}^k = \frac{\sum_{(u,\hat{u})\in TP} IoU(u,\hat{u})}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \quad (1)$$

As for video panoptic quality [9], a final VSQ score is computed by averaging VSQ^k over a set of windows, K:

$$\text{VSQ} = \frac{1}{K}\sum_k \text{VSQ}^k \quad (2)$$

For all experiments in this paper, $K = \{1, 5, 10, 15\}$ and VSQ^k is calculated using a window stride of 15 frames. To enable evaluation of frame-level models by VSQ, models may either directly output tubes or have output tubes formatted from frame-level predictions before evaluation. To format tubes from frame-level predictions, the Hungarian algorithm is applied in a pairwise fashion over their frame-level output segments. For models in this paper's experiments, Video K-Net and Tube-Link output tubes directly while SAM, FastSAM and FastSPAM have tubes formatted as described.

V. FastSPAM: Self-Prompting Anything Model

Building on recent foundation models for promptable segmentation [3], [4], we develop a self-prompting mechanism to enhance FastSAM [4] for greater accuracy and consistency in class-agnostic video instance segmentation. The key insight for using self-prompting lies in the hypothesis that promptable segmentation models like FastSAM can be made to output segments with reduced flickering if the prompts fed as input are informed by 3D spatial cues. To investigate this hypothesis, spatio-temporal self-prompting was developed to ensure the set of input prompts remain grounded in 3D space regardless of changes in camera viewpoint, thereby aiming to reduce the amount of flickering in the model's output over sequential frames. The resulting Fast Self-Prompting Anything model is referred to as FastSPAM. For a visual illustration, readers are referred to the supplementary video.

Fast*SPAM* performs segmentation in two stages: (1) an all-instance stage, which detects and predicts segments given an image as input. (2) A prompt-guided selection stage, which uses the input image together with a prompt (i.e. a point coordinate) to refine the detected segments. FastSPAM uses a sequence of RGB-D images, $(I^{0:T}, D^{0:T})$, as well as the corresponding camera projection matrices, $C^{0:T}$, as input. The camera projection matrix is defined to include both intrinsic and extrinsic parameters, $C = K\left[R|T\right]$, where K is the camera calibration matrix and $\left[R|T\right]$ is a homogeneous matrix defining the camera pose in world coordinate frame.

At each point in time FastSPAM maintains a set of self-prompts $P^t = \{p_0, \ldots, p_N\}^t$ where each self-prompt $p_i^t \in \mathbb{R}^3$ is a 3D point in the world coordinate frame describing the estimated centroid of each object. Given these pieces of information at time, $t-1$, FastSPAM first applies the YOLOv8-seg [53] method on the image I^t to perform all-instance segmentation. Next, the self-prompts are projected into the current image frame: $P' = \{C^t p_0^{t-1}, \ldots, C^t p_N^{t-1}\}$. Predicted segments from the all-instance stage are merged by union according to the self-prompts to ensure a single mask is predicted for each self-prompt. Finally for future predictions, an updated set of self-prompts, P^t, is calculated by converting (or 'unprojecting') the pixel-coordinate of each predicted segment's centroid into a 3D point in the world coordinate frame using the depth and camera matrix.

VI. Experiments

Dataset: MVPd is split into training, validation, and test subsets randomly at the scene-level (i.e. scenes in the training set have no videos represented in the test set). The held-out test set is used for all evaluations, while the training and validation sets are used for model learning and tuning.

Baseline Models: Two types of baseline models are considered: foundation models for image segmentation and fine-tuned models for video instance segmentation. The specific foundation models used for this paper include Segment Anything Model (SAM) [3] and FastSAM [4]. Two variants of SAM are used: SAM-B which uses ViT-B as its backbone and SAM-H which uses ViT-H as its backbone. In contrast, Video K-Net [17], Tube-Link [20], and OV2Seg [35] are used as baselines that were developed for video instance segmentation. Two variants of each baseline are included in these experiments: one variant uses a ResNet50 backbone and another uses a Swin-base or Swin-large backbone.

Implementation Details: Pre-trained SAM models are evaluated in automatic mode using default settings (32x32 uniform grid of point prompts). Finetuned FastSAM and FastSPAM models are trained on a single RTX A6000 GPU using a batch size of 16 images for 169K iterations (1

MVPd epoch). All other hyperparemters of FastSAM are left unchanged. Video K-Net and Tube-Link are trained with a mini-batch of 1 sample per GPU and a frame-range of 5 images per sample. OV2Seg models (ResNet50, Swin-base) are trained with a batch size of 16 and 8 images respectively. Each baseline is trained in a distributed fashion with 8 Tesla V100-GPUs until convergence (200K iterations). Otherwise, the baseline models are trained using the respective authors' published implementations and hyperparameter and pre-training settings; we did not tune the hyperparameters used for these models except to increase the total training iterations to ensure model convergence on MVPd (200k iterations instead of 100k [17] and 6-8k [20]). All models are trained with instance labels.

A. Pre-trained Foundation Models

In the first experiment, we set out to understand the effectiveness of foundation models that were pre-trained on a large-scale class-agnostic image segmentation dataset (SA-1B [3]) when evaluated on MVPd and class-agnostic *video* instance segmentation. To address this question, pre-trained foundation models [3], [4] (ViT-H SAM, ViT-B SAM), and FastSAM) were applied to each image in MVPd's test set and evaluated using the VSQ metric (Sec. IV). Video K-Net and Tube-Link are excluded from this evaluation since the pre-trained parameters publicly available are not intended for zero-shot transfer. The quantitative results are included in Tab. II. Comparing each model on the VSQ metric suggests an inverse relationship between model complexity and VSQ score. Inspection suggests that SAM's VSQ score suffers due to SAM's frequent sub-part predictions, which are considered to be false-positives in this task. **These results indicate that directly transferring image-based foundation models to video instance segmentation, despite their large-scale pre-training dataset, results in limited segmentation quality.**

B. After Finetuning Models

Next, we set out to evaluate the extent to which MVPd can support improved segmentation quality via finetuning. FastSAM , Video K-Net, and Tube-Link were finetuned on the MVPd training set and evaluated on its test set. The SAM models are excluded from this experiment due to resource limitations and their substantial training GPU requirements.

Quantitative results are shown in Tab. III indicating that for each window setting considered, FastSAM achieved the highest VSQ^k as well as the highest VSQ score of 49.17%, which is +6.65%, +24.05%, and +10.98% higher than top-performing baseline models respectively. The relative VSQ performance difference between Video K-Net, OV2Seg, and Tube-Link may result from Tube-Link's more complicated attention-based linking. Qualitative examples comparing each model are included in Fig. 6. **These results suggest image-based foundation models which were pre-trained on large-scale image datasets can be competitive with models designed specifically for video segmentation, if given sufficient in-domain data for finetuning.**

Method	Backbone	VSQ^k with k-Frame Window				VSQ
		$k=1$	$k=5$	$k=10$	$k=15$	
SAM	ViT-H	29.78	19.33	13.38	10.27	18.19
SAM	ViT-B	32.48	20.48	13.99	10.70	19.41
FastSAM	YOLOv8	**41.02**	**31.13**	**24.19**	**20.01**	**29.09**

TABLE II: Evaluating pre-trained foundation models (SAM-H, SAM-B, FastSAM) on the class-agnostic video instance segmentation task. Evaluation performed on videos from the held-out test set of MVPd and the VSQ evaluation metric (Sec. IV).

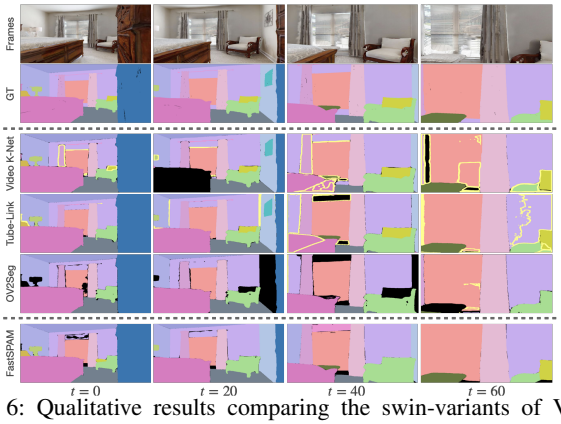

Fig. 6: Qualitative results comparing the swin-variants of Video K-Net, Tube-Link, OV2Seg, and FastSPAM. FastSPAM exhibits segments that are more consistent and accurate than the baselines and with fewer false positive predictions (outlined in neon yellow).

C. Using Depth and Camera Pose Modalities

We next set out to understand whether 3D modalities can be used to improve model segmentation quality. To answer this question, a self-prompting mechanism and FastSPAM model (Sec. V) were developed that use camera pose and depth (which FastSAM does not use) to recursively generate the prompts used to create its segment predictions.

Quantitative results, both in the pre-trained and finetuned settings, are shown in Tab. IV. Incorporating self-prompting resulted in an improvement of +2.55% VSQ compared to FastSAM in the pre-trained setting and +2.73% after finetuning. Moreover, FastSPAM achieved higher VSQ than both pre-trained SAM models (Tab. II) and both finetuned baselines (Tab. III). Qualitative examples are included in Fig. 7. **These results indicate that depth and camera pose features are useful features for temporally consistent segmentation and suggest future directions that make full use of these modalities during training.**

D. Controlling for Sensor Placement

This experiment set out to quantify how class-agnostic video instance segmentation models are impacted by a robot's embodiment. To carry out this experiment, we used

Method	Backbone	VSQ^k with k-Frame Window				VSQ
		$k=1$	$k=5$	$k=10$	$k=15$	
Video K-Net	ResNet50	49.65	49.60	38.10	29.56	41.73
Video K-Net	Swin-base	50.83	50.50	38.71	30.05	42.52
Tube-Link	ResNet50	45.13	20.50	16.08	14.05	23.94
Tube-Link	Swin-large	48.17	21.14	16.60	14.55	25.12
OV2Seg	ResNet50	39.96	38.52	37.51	36.71	38.18
OV2Seg	Swin-base	40.22	38.58	37.43	36.52	38.19
FastSAM	YOLOv8	**61.18**	**52.03**	**44.38**	**39.09**	**49.17**

TABLE III: Evaluating the impact of finetuning on model performance under VSQ metric and the MVPd test set.

Method	Training	VSQ^k with k-Frame Window				VSQ
		$k=1$	$k=5$	$k=10$	$k=15$	
FastSAM	PT (SA-1B)	41.02	31.13	24.19	20.01	29.09
FastSPAM	PT (SA-1B)	**42.95**	**33.65**	**27.05**	**22.89**	**31.64**
FastSAM	FT (MVPd)	**61.18**	52.03	44.38	39.09	49.17
FastSPAM	FT (MVPd)	60.68	**54.10**	**48.48**	**44.35**	**51.90**

TABLE IV: Evaluating the impact of using depth and camera pose to form self-prompts and improve FastSAM's performance under VSQ and the MVPd test set. Models with 'PT' were pretrained on SA-1B [3] while 'FT' models were finetuned on MVPd.

the sensor placement control data in MVPd to evaluate the top-performing algorithms' VSQ performance as a function of camera height above the floor. The pre-trained and fine-tuned FastSAM and FastSPAM models were evaluated on MVPd's test set videos, in which each video trajectory was recorded once at 1m height and a second time at 0.1m height. Thus, sensor height is the controlled variable.

Quantitative results are shown in Tab. V indicating that in both settings and for both FastSAM and FastSPAM, lower sensor placement is consistently associated with reduced video segmentation quality. This observed relationship may be a consequence of the distribution of camera perspectives represented in the pre-training dataset (SA-1B [3]), which was captured by human photographers whose height is likely closer to 1m than 0.1m. These results show that training on MVPd reduces the gap in performance associated with sensor height: After finetuning on MVPd, FastSAM's performance gap as a function of sensor placement reduces from 2.79% to 2.78% VSQ. The gap is further reduced by both finetuning and using self-prompting as shown by FastSPAM's gap reducing from 4.21% to 1.09% VSQ. **These results suggest two directions for future work to improve video segmentation quality for embodied robots. First, temporal aggregation strategies beyond self-prompting may be useful for both models trained from supervision and finetuned models. Second, for embodied robots seeking to use video segmentation models trained from supervision, having access to data that represents the robot's morphology is beneficial and motivates using a data generation pipeline like the one used for MVPd to create embodiment-specific training data.**

E. Results on Zero-Shot Subset

Next, models are compared based on their accuracy and consistency using MVPd's zero-shot subset (Sec. III-B).

Method	Training	Sensor Placement	VSQ^k with k-Frame Window				VSQ	ΔVSQ
			$k=1$	$k=5$	$k=10$	$k=15$		
FastSAM	PT	1m	42.29	32.43	25.42	21.13	30.32	2.79
		0.1m	39.37	29.47	22.65	18.62	27.53	
FastSPAM	PT	1m	**44.88**	**35.60**	**28.93**	**24.66**	**33.52**	4.21
		0.1m	40.49	31.21	24.76	20.76	29.31	
FastSAM	FT	1m	**62.36**	53.31	45.66	40.35	50.42	2.78
		0.1m	59.71	50.46	42.82	37.57	47.64	
FastSPAM	FT	1m	61.14	**54.58**	**48.99**	**44.87**	**52.39**	1.09
		0.1m	60.10	53.51	47.85	43.73	51.30	

TABLE V: Evaluating the impact of sensor placement on FastSAM and FastSPAM VSQ on the MVPd test set. Models with 'PT' were pretrained on SA-1B [3] while 'FT' models were finetuned on MVPd. ΔVSQ measures difference in model's VSQ between evaluation videos at 1m and 0.1m.

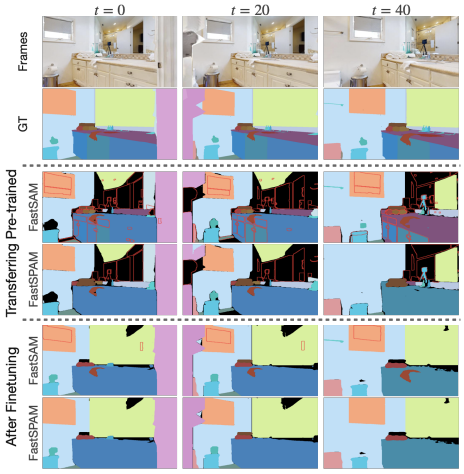

Fig. 7: Qualitative comparison of FastSAM and FastSPAM. Middle panel: predictions before domain-specific finetuning. Right panel: predictions after finetuning with MVPd. FastSPAM exhibits reduced false positives as a result of self-prompting. Predicted segments are colored according to an optimal match against the ground truth segments based on pairwise F-score (Sec. IV): predicted segments matching a ground truth are assigned the corresponding color while unmatched predictions (false positives) are denoted by a red outline.

Method	Backbone	VSQ^k with k-Frame Window				VSQ
		$k=1$	$k=5$	$k=10$	$k=15$	
Video K-Net	ResNet50	4.52	4.82	4.00	3.29	4.16
Video K-Net	Swin-base	5.22	5.63	4.78	3.92	4.89
Tube-Link	ResNet50	2.28	1.03	0.82	0.73	1.22
Tube-Link	Swin-large	2.76	1.26	1.02	0.94	1.49
OV2Seg	ResNet50	4.46	5.45	6.21	6.83	5.74
OV2Seg	Swin-base	5.37	6.65	7.59	8.33	6.99
FastSAM	YOLOv8	14.38	6.50	3.49	2.36	6.68
FastSPAM	YOLOv8	**22.51**	**11.96**	**6.57**	**4.43**	**11.37**

TABLE VI: Evaluating finetuned models on the zero-shot subset of MVPd. All objects in this evaluation belong to categories and videos that were never seen during training.

This comparison aims to establish a measure of accuracy to expect of models deployed in the 'open-world' where they encounter categories of objects not seen at training. Quantitative results are shown in Tab. VI indicating FastSPAM outperforms the top-performing baseline models as well as FastSAM on zero-shot objects by substantial margins of +6.48%, +9.88%, +4.38%, and +4.69% VSQ respectively. **These results together with those in Sec. VI-C indicate that self-prompting is beneficial for improving model performance on out-of-training-distribution data.**

VII. Conclusion

This paper makes three central contributions: (1) the introduction of a massive RGB-D video segmentation dataset and associated pipeline to support research on embodied video segmentation, (2) extensive benchmarking experiments that establish expected performance on class-agnostic video segmentation by state-of-the-art models, and (3) ablation experiments that demonstrate depth and camera pose modalities can benefit video segmentation accuracy and consistency. Specifically, the experiments demonstrated that incorporating spatio-temporal self-prompting (FastSPAM) with a foundation segmentation model (FastSAM) led to reduced segment

inconsistency and increased model accuracy. Furthermore, the explicit self-prompting mechanism is beneficial when applied to out-of-distribution data. Results from this study suggest future directions to further improve video segmentation models by incorporating self-prompting, depth, and camera pose during training. Furthermore, the data contribution may enable new research directions in embodied class-agnostic video segmentation for robotic use cases.

REFERENCES

[1] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva *et al.*, "Habitat-matterport 3d semantics dataset," in *CVPR*, 2023.

[2] D. Maggio, Y. Chang, N. Hughes, M. Trang, D. Griffith, C. Dougherty, E. Cristofalo, L. Schmid, and L. Carlone, "Clio: Real-time task-driven open-set 3d scene graphs," *arXiv:2404.13696*, 2024.

[3] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[4] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv:2306.12156*, 2023.

[5] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *ECCV*, 2014.

[6] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: state of the art," *IJMIR*, 2020.

[7] X. Ren and J. Malik, "Tracking as repeated figure/ground segmentation," in *CVPR*, 2007.

[8] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *ICCV*, 2019.

[9] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Video panoptic segmentation," in *CVPR*, 2020.

[10] T. Zhou, F. Porikli, D. J. Crandall, L. Van Gool, and W. Wang, "A survey on deep learning technique for video segmentation," *IEEE PAMI*, 2023.

[11] J. Miao, X. Wang, Y. Wu, W. Li, X. Zhang, Y. Wei, and Y. Yang, "Large-scale video panoptic segmentation in the wild: A benchmark," in *CVPR*, 2022.

[12] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *CVPR*, 2019.

[13] R. Benenson, S. Popov, and V. Ferrari, "Large-scale interactive object segmentation with human annotators," in *CVPR*, 2019.

[14] J. Xiao, A. Owens, and A. Torralba, "SUN3D: A database of big spaces reconstructed using sfm and object labels," in *ICCV*, 2013.

[15] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *3DV*, 2017.

[16] J. Vertens, A. Valada, and W. Burgard, "Smsnet: Semantic motion segmentation using deep convolutional neural networks," in *IROS*, 2017.

[17] X. Li, W. Zhang, J. Pang, K. Chen, G. Cheng, Y. Tong, and C. C. Loy, "Video k-net: A simple, strong, and unified baseline for video segmentation," in *CVPR*, 2022.

[18] S. Yang, X. Wang, Y. Li, Y. Fang, J. Fang, Liu, X. Zhao, and Y. Shan, "Temporally efficient vision transformer for video instance segmentation," in *CVPR*, 2022.

[19] Y. Xu, Z. Yang, and Y. Yang, "Video object segmentation in panoptic wild scenes," in *IJCAI-23*, 2023.

[20] X. Li, H. Yuan, W. Zhang, G. Cheng, J. Pang, and C. C. Loy, "Tube-link: A flexible cross tube framework for universal video segmentation," in *ICCV*, 2023.

[21] M. Siam, A. Kendall, and M. Jagersand, "Video class agnostic segmentation benchmark for autonomous driving," in *CVPRW*, 2021.

[22] P. Tokmakov, K. Alahari, and C. Schmid, "Learning motion patterns in videos," in *CVPR*, 2017.

[23] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *CVPR*, 2017.

[24] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *CVPR*, 2017.

[25] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *ICCV*, 2017.

[26] X. Wang, X. Zhang, Y. Cao, W. Wang, C. Shen, and T. Huang, "Seggpt: Towards segmenting everything in context," in *ICCV*, 2023.

[27] M. Bucher, T.-H. VU, M. Cord, and P. Pérez, "Zero-shot semantic segmentation," in *NeurIPS*, 2019.

[28] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data," in *IEEE ICRA*, 2019.

[29] Y. Zheng, J. Wu, Y. Qin, F. Zhang, and L. Cui, "Zero-shot instance segmentation," in *CVPR*, 2021.

[30] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, "Learning rgb-d feature embeddings for unseen object instance segmentation," in *CoRL*, 2021.

[31] E. P. Örnek, A. K. Krishnan, S. Gayaka, C.-H. Kuo, A. Sen, N. Navab, and F. Tombari, "Supergb-d: Zero-shot instance segmentation in cluttered indoor environments," *IEEE R-AL*, 2023.

[32] Y. Du, Y. Xiao, and V. Lepetit, "Learning to better segment objects from unseen classes with unlabeled videos," in *ICCV*, 2021.

[33] I. Nunes, C. Laranjeira, H. Oliveira, and J. A. dos Santos, "A systematic review on open-set segmentation," *Comp. & Grphcs.*, 2023.

[34] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee, "Tracking anything with decoupled video segmentation," in *ICCV*, 2023.

[35] H. Wang, C. Yan, S. Wang, X. Jiang, X. Tang, Y. Hu, W. Xie, and E. Gavves, "Towards open-vocabulary video instance segmentation," in *ICCV*, 2023.

[36] H. Wang, C. Yan, K. Chen, X. Jiang, X. Tang, Y. Hu, G. Kang, W. Xie, and E. Gavves, "Ov-vis: Open-vocabulary video instance segmentation," *International Journal of Computer Vision*, pp. 1–18, 2024.

[37] X. Li, H. Yuan, W. Li, H. Ding, S. Wu, W. Zhang, Y. Li, K. Chen, and C. C. Loy, "Omg-seg: Is one model good enough for all segmentation?" in *CVPR*, 2024.

[38] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021.

[39] J. Cen, Z. Zhou, J. Fang, C. Yang, W. Shen, L. Xie, X. Zhang, and Q. Tian, "Segment anything in 3d with nerfs," in *NeurIPS*, 2023.

[40] M. Weber, J. Xie, M. Collins, Y. Zhu, P. Voigtlaender, H. Adam, B. Green, A. Geiger, B. Leibe, D. Cremers, A. Osep, L. Leal-Taixe, and L.-C. Chen, "Step: Segmenting and tracking every pixel," in *NeurIPS*, 2021.

[41] L. Yang, Y. Fan, and N. Xu, "The 4th large-scale video object segmentation challenge - video instance segmentation track," 2022.

[42] J. Qi, Y. Gao, Y. Hu, X. Wang, X. Liu, X. Bai, S. Belongie, A. Yuille, P. Torr, and S. Bai, "Occluded video instance segmentation: A benchmark," *IJCV*, 2022.

[43] W. Wang, M. Feiszli, H. Wang, and D. Tran, "Unidentified video objects: A benchmark for dense, open-world segmentation," in *ICCV*, 2021.

[44] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[45] J. Miao, Y. Wei, Y. Wu, C. Liang, G. Li, and Y. Yang, "Vspw: A large-scale dataset for video scene parsing in the wild," in *CVPR*, 2021.

[46] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *CVPR*, 2017.

[47] A. Eftekhar, A. Sax, J. Malik, and A. Zamir, "Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans," in *ICCV*, 2021.

[48] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *ICCV*, 2019.

[49] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection," in *ECCV*, 2018.

[50] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[51] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *CVPR*, 2019.

[52] A. Dave, P. Tokmakov, and D. Ramanan, "Towards segmenting everything that moves," *arXiv:1902.03715*, 2019.

[53] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023.