SegMASt3R: Geometry Grounded Segment Matching

Rohit Jayanti[1*†] **Swayam Agrawal**[1*‡] **Vansh Garg**[1*] **Siddharth Tourani**[2,3]
Muhammad Haris Khan[3] **Sourav Garg**[4] **Madhava Krishna**[1]

[1]IIIT Hyderabad [2]University of Heidelberg [3]MBZUAI [4]Independent

Project Page: `https://segmast3r.github.io/`

Abstract

Segment matching is an important intermediate task in computer vision that establishes correspondences between semantically or geometrically coherent regions across images. Unlike keypoint matching, which focuses on localized features, segment matching captures structured regions, offering greater robustness to occlusions, lighting variations, and viewpoint changes. In this paper, we leverage the spatial understanding of 3D foundation models to tackle wide-baseline segment matching, a challenging setting involving extreme viewpoint shifts. We propose an architecture that uses the inductive bias of these 3D foundation models to match segments across image pairs with up to 180° rotation. Extensive experiments show that our approach outperforms state-of-the-art methods, including the SAM2 video propagator and local feature matching methods, by up to 30% on the AUPRC metric, on ScanNet++ and Replica datasets. We further demonstrate benefits of the proposed model on relevant downstream tasks, including 3D instance mapping and object-relative navigation.

1 Introduction

Segment matching establishes correspondences between coherent regions—objects, parts, or semantic segments—across images. It underpins video object tracking [17], scene-graph construction [18, 23], robot navigation [16, 15, 32], and instance-level SLAM [26, 44]. Because it matches extended structures rather than sparse, texture-sensitive points, it is more robust to noise, occlusion, and appearance change. Mapping structured regions instead of isolated pixels also boosts interpretability and allows geometric or semantic priors to be injected, enabling higher-level reasoning about scene content.

Segment matching degrades sharply under *wide-baseline* conditions, where images of the same scene are taken from widely separated viewpoints that introduce severe perspective, scale, and up to 180° rotation changes [34]. Such cases arise in long-term video correspondence and robotic navigation, and demand models that reason over global 3D structure and enforce geometric consistency. Current approaches that depend on features from pre-trained encoders like DINOv2 [30] or ViT [10] often mismatch repetitive patterns or fail to link drastically different views of the same object.

In this paper, we propose to leverage the strong spatial inductive bias of a 3D foundation model (namely MASt3R [22]) to solve the problem of wide-baseline segment matching. 3D Foundation Models (3DFMs) are large-scale vision models trained to capture spatial and structural properties of scenes, such as depth, shape, and pose. Unlike appearance-focused models, 3DFMs learn geometry-aware representations using 2D and 3D supervision, enabling strong generalization across tasks

[*] Equal Contribution. [†] Corresponding Author. [‡] Now at Google DeepMind.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

like reconstruction and pose estimation. Their inductive bias toward spatial reasoning makes them well-suited for applications requiring geometric consistency such as our problem of wide-baseline segment matching.

We adapt MASt3R for segment matching by appending a lightweight *segment-feature head* that transforms patch-level embeddings into segment-level descriptors. Given an image pair, these descriptors are matched to establish segment correspondences. The head is trained with a contrastive objective patterned after SuperGlue [35]. Experiments show that our approach surpasses strong baselines, including SAM2's video propagator which is trained on far larger datasets and state-of-the-art local feature matching methods. Finally, we demonstrate its practical utility in two downstream tasks: object-relative navigation and instance-level mapping.

Contributions Our key contributions are summarized below:

- We introduce a simple but effective approach for learning segment-aligned features by leveraging strong priors from a 3D foundation model (3DFM) MASt3R. A differentiable segment matching layer is employed to align features across views, while a *segment-feature head* transforms dense pixel-level representations into robust segment-level descriptors.
- To address the under-explored problem of wide-baseline segment matching, we construct a comprehensive benchmark comprising both direct segment association methods and those based on local feature matching. Our method demonstrates significant improvements over all baselines on challenging wide-baseline image pairs.
- We validate the practical utility of our approach by applying it to the downstream applications such as 3D instance mapping and object-relative topological navigation. Our method outperforms competitors by significant margins showcasing the efficacy of our design choices.

2 Related Work

Segment Matching and Segmentation Foundation Models. Robust segment-level association has emerged as a crucial intermediate step for high-level vision tasks such as scene graph construction, long-term object tracking, and multi-view instance association. While related sub-problems like video instance segmentation and object tracking have been studied extensively in recent works [46, 45, 41, 11, 31, 29, 24, 6], the broader challenge of matching segments across arbitrary viewpoints, modalities, and time remains comparatively under-explored. Large-scale class-agnostic segmentation models have begun to close this gap. The Segment Anything Model (SAM) [20] and its successor SAM2 [34] deliver high-quality masks and include a built-in propagator for associating masks across video frames. However, this propagation module is optimized for short temporal windows and does not explicitly enforce geometric consistency under wide baselines or substantial appearance changes.

Learning to Match Segments and Overlap Prediction. Some recent methods address segment matching more directly. MASA [25] augments SAM's rich object proposals with synthetic geometric transformations to learn instance correspondences, and DMESA [49] extends these ideas to dense matching with improved efficiency. Despite such progress, these approaches are still limited by 2D supervision. An alternative line of work predicts the degree of visual overlap between images [7, 13, 2]. By estimating shared content, these methods implicitly learn region correspondences, yet they also remain confined to 2D training signals.

Local Feature Matching. Sparse [9, 35, 40, 27, 36] and dense [12, 4] local feature matchers propagate pixel-level correspondences that can, in principle, transfer segment labels between views [15, 16]. Nonetheless, like the previous categories, they are trained exclusively on image data and struggle with extreme viewpoint changes.

Across segment matching, overlap prediction, and feature matching, reliance on purely 2D supervision leaves existing techniques brittle under wide-baseline conditions. Our method addresses this limitation by fine-tuning the 3D foundation model MASt3R [22], whose strong geometric priors enable reliable segment correspondence even when image pairs differ by nearly 180° in viewpoint.

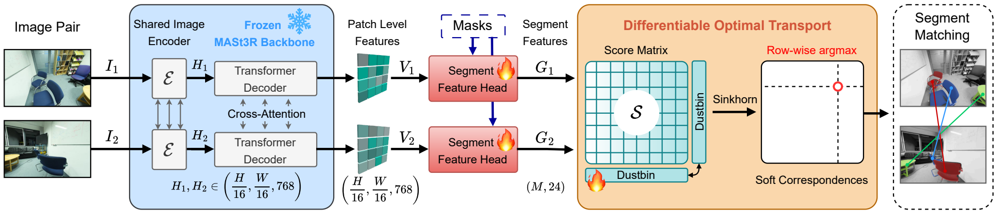


Figure 1: **Pipeline Overview**: An image pair is processed by a frozen MASt3R backbone to extract patch-level features; segmentation masks are obtained either from a parallel segmentation module or ground truth annotation; the patch-level features are aggregated by the segment-feature head to form segment-level descriptors; and these descriptors are then matched across images via a differentiable optimal transport layer to produce segment-level correspondences.

3 Method

Figure 1 provides an overview of our method. It builds upon the MASt3R [22] architecture by introducing a Feat2Seg Adapter that maps the patch-level features output by the MASt3R decoder to get segment-level features, which are subsequently matched via a differentiable optimal transport and a row-wise argmax yielding the final segment-level correspondences.

3.1 MASt3R Preliminaries

MASt3R is a 3D foundation model pre-trained on a diverse collection of 3D-vision datasets commonly used for tasks such as metric depth estimation and camera-pose prediction. Given a pair of images, it produces dense 3D point maps for each image and identifies correspondences between them. Thanks to training on data with wide-baseline pairs, MASt3R generalizes well to unseen image pairs [22] and consistently outperforms alternative 3D matching methods—such as MicKey [3], which relies on a DINOv2 backbone [3].

Architecture: We summarize here the portions of the architecture we utilize in our pipeline. The two images I_1 and I_2 are processed in a Siamese manner by a weight-sharing ViT encoder [10] $\mathcal{E}$, resulting in token sets H_1, H_2, *i.e.*

$$H_1 = \mathcal{E}(I_1), \qquad H_2 = \mathcal{E}(I_2).$$

Cross-view transformer decoder: Next, a pair of CroCo [42, 43] style intertwined transformer decoders jointly refines the two feature sets. By alternating self- and cross-attention, the decoders exchange information between inputs to capture both the relative viewpoints and the global 3-D structure of the scene. We show via an ablation study, these cross-view aware decoders provide a significant boost to the segment matching in Section 5

The resulting geometry-aware representations are denoted V_1 and V_2:

$$\big(V_1,\ V_2\big) = \text{Decoder}\big(\mathbf{H}_1,\ \mathbf{H}_2\big).$$

Assuming, H, W denote the height and width of the input images, the output geometry-aware patch-level features are of size $(H/16, W/16, 768)$. The original architecture subsequently uses two prediction heads to output dense point-maps, as well as pixel-level features, they are not shown in Figure 1 as they are not used in our pipeline. These portions of the pipeline remain frozen and their outputs are used as is, we instead introduce a new segment-feature head.

3.2 Segment-Feature Head: Segment-Aligned Features

The MASt3R decoder produces patch–level embeddings V_1, V_2 of shape $(H/16, W/16, 768)$. In the original MASt3R [22], a *feature head* upsamples these tensors to the resolution of the input image. For *segment matching* we introduce another head to transform the patch level features to M segment features. This introduced head is realized as an MLP that upsamples the patch-level features (V_1, V_2) to image resolution, yielding feature maps of size $(H, W, 24)$, 24 being the feature dimension. We denote this *Feature-to-Segment* head as the **segment-feature** head. In addition to the patch-features, the **segment-feature** takes as input M image resolution segment masks, obtained either from an external segmenter such as SAM2 [34] or from ground-truth annotations. Both the masks and feature maps are flattened along the spatial dimensions yielding flattened tensors.

$$\mathbf{P}_{\text{flat}} \in \mathbb{R}^{24 \times HW}, \qquad \mathbf{M}_{\text{flat}} \in \mathbb{R}^{M \times HW}.$$

To go from pixel-level descriptors to segment descriptors, a single batched matrix multiplication aggregates the pixel descriptors inside each mask:

$$\mathbf{G} = \mathbf{M}_{\text{flat}}\ \mathbf{P}_{\text{flat}}^{\top} \in \mathbb{R}^{M \times 24}. \tag{1}$$

The resulting segment embeddings for the two images are denoted $\mathbf{G}_1$ and $\mathbf{G}_2$ in Figure 1 and are fed to the differentiable matching layer described in Section 3.3. We set $M = 100$ as an upper bound for batch processing. In practice, images typically contain 20-30 masks when training with ground truth annotations; we pad with zeros when fewer masks are present. At inference time, the number of masks can be set arbitrarily, independent of the training-time value of M.

3.3 Differentiable Segment Matching Layer

The goal of the differentiable segment matching layer is to establish permutation-style correspondences between the M segments from each image, given segment descriptors $(G_1, G_2) \in \mathbb{R}^{(M,24)}$

Cosine–similarity affinity. We first construct an affinity matrix $\mathbf{S} \in \mathbb{R}^{M_1 \times M_2}$ with a simple dot product

$$S_{ij} = \langle \mathbf{g}_i^1, \mathbf{g}_j^2 \rangle, \quad 1 \le i \le M_1,\ 1 \le j \le M_2. \tag{2}$$

$\mathbf{g}_i^1$ and $\mathbf{g}_j^2$ correspond to segment-level features from G_1 and G_2 respectively. Ideally, segment features corresponding to the same underlying 3D region should have a high similarity score and dis-similar regions a correspondingly low score.

Learnable dustbin. Following [35], we incorporate a *dustbin* row and column in the affinity matrix $\mathbf{S}$ to handle segments without correspondences in the other image, which is critical for wide-baseline matching. We augment (2) by concatenating an additional row and column initialized with a learnable logit $\alpha \in \mathbb{R}$, yielding $\tilde{\mathbf{S}} \in \mathbb{R}^{(M_1+1)\times(M_2+1)}$.

$$\tilde{\mathbf{S}} = \begin{bmatrix} \mathbf{S} & \alpha \mathbf{1}_{M_1} \\ \alpha \mathbf{1}_{M_2}^{\top} & \alpha \end{bmatrix},$$

Soft Correspondences via Sinkhorn The similarity logits are transformed into a soft assignment matrix $\mathbf{P}$ by T iterations of the Sinkhorn normalisation [38, 37] in log-space:

$$\begin{aligned} \mathbf{P}^{(0)} &\leftarrow \exp\bigl(\tilde{\mathbf{S}}/\tau\bigr), \\ u_i^{(t)} &= \frac{1}{\sum_j P_{ij}^{(t)}}, \quad v_j^{(t)} = \frac{1}{\sum_i P_{ij}^{(t)}}, \\ P_{ij}^{(t+1)} &= u_i^{(t)}\, P_{ij}^{(t)}\, v_j^{(t)}, \quad 0 \le t < T, \end{aligned} \tag{3}$$

where τ is a temperature hyper-parameter. After convergence, $\mathbf{P} = \mathbf{P}^{(T)}$ is (approximately) doubly stochastic. Throughout all experiments conducted, we assume $T = 50$. The output of the Sinkhorn algorithm $\mathbf{P}^{(T)}$ is a soft bi-stochastic matrix, which has to be discretized to obtain the final segment matches.

Discrete correspondences. To obtain the simple final segment matches, we perform a simple row-wise $\arg\max$ over the *non-dustbin* columns:

$$m(i) = \operatorname*{argmax}_{1 \le j \le M_2} P_{ij}, \quad \text{with assignment accepted if } j \ne M + 1.$$

3.4 Supervision

Training objective. We adopt the SuperGlue cross-entropy loss $\mathcal{L}_{\text{SG}}$ [35], extended with explicit terms for unmatched segments:

$$\mathcal{L} = -\sum_{(i,j)\in\mathcal{M}} \log P_{ij} - \sum_{i\in\mathcal{U}_1} \log P_{i,M+1} - \sum_{j\in\mathcal{U}_2} \log P_{M+1,j}, \tag{4}$$

where $\mathcal{M}$ is the set of ground-truth matches and $\mathcal{U}_1, \mathcal{U}_2$ are the unmatched indices in image 1 and 2, respectively. The dustbin parameter α is learned jointly with the rest of the network, enabling the layer to balance match confidence against the cost of declaring non-matches. This fully differentiable design allows the matching layer to be trained end-to-end together with the upstream segment encoders and downstream task losses.

3.4.1 Training Details

The model is trained using AdamW optimizer with an initial learning rate of `1e-4`, weight decay of `1e-4`, and a cosine annealing learning rate schedule without restarts, decaying up to a minimum learning rate of `1e-6` over the full training duration. We use a batch size of 36 and train the model for 20 epochs on a single NVIDIA RTX A6000 GPU. The **segment-feature heads** are initialized with MASt3R's local feature head weights and finetuned further. For the differentiable segment matcher we initialize the single learnable dustbin parameter to 1.0. The number of Sinkhorn iterations is set to 50. Training our model on ScanNet++ takes 22 hours, whereas a single forward pass during inference with batch size of 1 takes 0.579 seconds.

4 Experiments

4.1 Datasets

Our network is trained on scenes from ScanNet++ [47] which contain a diverse set of real-world scenes, primarily in indoor settings. We test our model on novel scenes from ScanNet++ as well as perform cross-dataset generalization studies on Replica [39] and MapFree [2] datasets. The former contains high-quality photo-realistic indoor scenes while the latter is a challenging outdoor visual-localization dataset, which is sufficiently out-of-distribution considering our training data.

ScanNet++ [47]. ScanNet++ contains 1 006 indoor scenes captured with DSLR images and RGB-D iPhone streams, all co-registered to high-quality laser scans. The dataset supplies 3D semantic meshes, 2D instance masks, and accurate camera poses, which we utilize in our pipeline. We train on 860 k image pairs from 140 scenes and evaluate on 8 k pairs from 36 validation scenes, sampled with a fixed seed (42) and balanced across scenes and four pose bins: $[0° - 45°]$, $[45° - 90°]$, $[90° - 135°]$, and $[135° - 180°]$, defined by the rotational geodesic distance between camera orientations.

Replica [39]. The Replica dataset contain 18 high-quality, photo-realistic indoor room reconstructions in the form of dense meshes replete with per-primitive semantic class and instance information. In particular, we use a pre-rendered version of this dataset from Semantic-NeRF [50], which directly provides poses, RGB-D sequences, and per-frame semantic masks. For evaluation, we employ the same pose-binning strategy as described above and randomly sample 3200 image pairs across 8 scenes, again ensuring uniform sampling across both scenes and pose-bins.

MapFree Visual Re-localization [2]. MapFree is a challenging benchmark for metric-relative pose estimation, featuring 655 diverse outdoor scenes (sculptures, fountains, murals) with extreme viewpoint changes, varying visual conditions, and geometric ambiguities. The training split contains 460 scenes with 0.5M images; we uniformly sample 50 scenes yielding 31K image pairs for training. Since test split ground truth is unavailable, we report results on the validation split using 7.8K pairs from 13 uniformly sampled scenes (out of 65 total).

4.2 Baselines

Local Feature Matching (LFM). Although few models target *segment* matching directly, a wide range of local feature matchers (LFMs) exists at different densities: sparse SuperPoint [9], semi-dense LoFTR [40], and dense RoMa and MASt3R [12, 22]. We harness these state-of-the-art LFMs

Type	Method	0°–45°			45°–90°			90°–135°			135°–180°		
		AUPRC	R@1	R@5	AUPRC	R@1	R@5	AUPRC	R@1	R@5	AUPRC	R@1	R@5
Local Feature Matching (LFM)	SP-LG [9, 27]	42.1	45.6	51.2	33.5	36.9	43.1	15.9	19.7	26.2	6.1	9.3	14.6
	GiM-DKM [36, 11]	59.1	64.9	69.7	54.9	60.2	66.1	39.6	44.5	51.8	21.3	25.9	32.7
	RoMA [12]	61.6	68.7	73.5	58.9	66.4	73.0	47.4	56.1	65.5	30.0	39.5	49.7
	MASt3R [22]	59.5	68.3	74.2	57.3	65.6	72.5	52.9	60.3	68.9	45.4	52.6	62.2
Segment Matching (SegMatch)	SAM2 [34]	61.9	64.6	67.5	46.6	50.1	54.0	27.9	32.5	37.2	17.0	21.6	25.4
	DINOv2 [30]	57.9	66.7	87.4	43.0	55.9	83.2	33.5	48.0	78.0	32.4	46.0	75.6
	SegVLAD [14]	44.2	58.6	81.4	32.1	49.5	76.5	23.2	42.2	70.5	20.0	39.6	66.8
	MASt3R [22]	51.7	54.6	69.9	45.6	49.8	68.5	41.4	47.9	69.2	39.5	48.7	72.6
Ours	SEGMAST3R	92.8	93.6	98.0	91.1	92.2	97.6	88.0	89.5	96.8	83.6	85.9	95.9

Table 1: Performance of selected methods across pose-bins on ScanNet++ [47]. Blue cells mark the best scores; Orange cells mark the second-best.

via the EarthMatch toolkit [5] to obtain segment correspondences. Each matched keypoint pair votes for the source and target masks that contain its coordinates, populating a vote matrix of size $M \times N$ (with M and N segments in the two images). Correspondences are taken as the highest-scoring entries of this matrix; the full algorithm is provided in the supplementary.

Segment Matching (SegMatch). We also compute segment matches from dense features of two strong pre-trained vision encoders, DINOv2 [30] and MASt3R [22], as well as SAM2's video propagator [34] to track masks across views. For feature based methods, masks are downsampled via nearest-neighbor interpolation to match feature resolution, and segment descriptors are computed via masked average pooling. Cosine similarity between descriptors yields a match matrix, from which one-to-one links are selected via mutual-check. We further benchmark against SegVLAD [14], which aggregates features from neighbouring segments for segment retrieval based visual place recognition.

4.3 Evaluation Metrics

We report two complementary measures of segment correspondence quality - **AUPRC** and **Recall**. *Area Under the Precision–Recall Curve (AUPRC)* integrates precision over the entire recall axis, providing a threshold–independent summary that is particularly informative under the high class–imbalance characteristic of segment matching between image pairs. *Recall*@k ($R@k$) denotes the fraction of query segments whose ground-truth counterpart is found within the top k ranked candidates, thus gauging how effectively the method surfaces correct matches among its highest-confidence predictions. Additional details for the dataset, baseline and experiments can be found in the supplementary, along with more qualitative results.

5 Results

Segment Matching In Table 1, we compare our proposed method SegMASt3R with the state-of-the-art methods using two categories of approaches in the literature: the well-established local feature matching (LFM) based on sparse or dense keypoint correspondences, and the recently emerging open-set instance association based on segment matching (SegMatch). It can be observed that SegMASt3R outperforms all the baselines for all the pose bins with a huge margin. Amongst the LFM techniques, dense matchers (RoMa and MASt3R) outperform sparse matchers (SP-LG and GiM-DKM) on both the evaluation metrics for the task of segment matching. Notably, on the highly challenging wide-baseline settings, MASt3R outperforms other local matchers by a large margin. However, when the same backend features are aggregated at the segment-level, MASt3R's performance deteriorates significantly, e.g., AUPRC drops from 52.9 to 41.4 on the 90-135 pose bin. This uncovers an inherent limitation: *feature distinctiveness at the pixel level does not necessarily translate to the instance level*. Considering the SegMatch techniques which are not trained for the LFM task, it can be observed that DINOv2 and SegVLAD perform particulary well on R@5 metric, which aligns with their typical use for coarse retrieval [19, 14]. On the other hand, SAM2's two-frame video propagation only works well for narrow-baseline matching, which can be expected as its training set is mostly comprised of dynamic object tracking. Overall, these results show that all

Type	Method	0°–45°			45°–90°			90°–135°			135°–180°		
		AUPRC	R@1	R@5	AUPRC	R@1	R@5	AUPRC	R@1	R@5	AUPRC	R@1	R@5
LFM	MASt3R [22]	78.2	86.5	89.4	69.5	77.6	81.0	48.0	60.4	64.6	32.5	49.0	54.1
SegMatch	MASt3R [22]	52.2	57.5	81.2	39.1	51.0	78.6	23.6	45.9	77.2	17.2	43.8	75.7
Ours	SEGMAST3R	95.0	96.0	98.6	86.2	91.2	96.4	73.4	85.2	95.7	68.4	83.8	94.8

Table 2: Performance of selected methods across pose-bins on Replica [39]. Blue cells mark the best scores; Orange cells mark the second-best.

the baseline methods lack on at least one of the fronts: pixel- vs segment-level distinctiveness, recall vs precision, and narrow- vs wide-baseline robustness. Our proposed method SegMASt3R achieves all these desirable properties with high performance across the board by learning segment-level representations with the training objective of instance association. In Section 5, we provide qualitative results which emphasize the ability of our method to address the problem of perceptual instance aliasing and instance matching under extreme viewpoint shifts.

Generalization We assess our model's ability to generalize to new environments. First, in Table 2, we present results on a different indoor dataset, Replica, to test the generalization ability of SegMASt3R, which is trained only on ScanNet++. We compare SegMASt3R against the LFM and SegMatch versions of MASt3R, as these three methods closely resemble each other in terms of their network architecture (detailed comparisons are included in the supplementary). It can be observed that the performance patterns on Replica remain largely the same as that on ScanNet++, and SegMASt3R consistently outperforms the LFM and SegMatch versions of MASt3R across the board.

Furthermore, we test generalization to challenging outdoor scenes from the MapFree dataset [2], with results shown in Table 3. Since MapFree lacks instance-level ground truth, we use SAM2's video propagator on image sequences to generate a pseudo-ground truth for evaluation. Our indoor-trained model (SegMASt3R (SPP)) shows a regression in performance in comparison to DINOv2, highlighting the domain shift. However, this gap can be substantially closed by either re-training on MapFree data (SegMASt3R (MF)) or, even more simply by just recalibrating the single learnable dustbin parameter α using a grid-search over a small calibration set from the target domain (SegMASt3R (SPP, Dustbin MF)). This demonstrates the strong adaptability of our model's learned geometric features.

Method	Train Set	Overall IoU	0-45°	45-90°	90-135°	135-180°
DINOv2 [30]	Multiple	84.4	85.4	85.2	83.5	83.8
MASt3R (Vanilla) [22]	Multiple	69.2	73.4	69.8	70.0	66.1
SegMASt3R (SPP)	ScanNet++	75.2	75.2	74.6	76.5	74.5
SegMASt3R (SPP, Dustbin MF)	ScanNet++	88.7	88.6	88.6	88.5	88.9
SegMASt3R (MF)	MapFree	93.7	93.3	93.7	93.9	93.9

Table 3: Generalization performance on the outdoor MapFree dataset [2].

Method	office0	office1	office2	office3	office4	room0	room1	room2
	AP / AP@50	AP / AP@50	AP / AP@50	AP / AP@50	AP / AP@50	AP / AP@50	AP / AP@50	AP / AP@50
ConceptGraphs (MobileSAM Masks) [17]	11.84 / 28.43	20.31 / 43.79	8.63 / 22.82	8.07 / 22.83	9.46 / 24.73	12.23 / 34.34	5.83 / 12.96	7.83 / 23.82
ConceptGraphs (GT Masks) [17]	43.53 / 69.68	22.48 / 40.71	43.46 / 60.69	32.06 / 53.44	39.63 / 68.22	44.89 / 69.64	17.96 / 36.53	25.93 / 43.63
SegMASt3R(Ours, GT Masks)	79.93 / 87.17	54.89 / 64.42	64.00 / 85.50	58.02 / 79.93	67.48 / 85.01	71.02 / 91.22	64.09 / 85.50	56.35 / 76.66

Table 4: Class-Agnostic instance-mapping performance (AP and AP@50) on Replica scenes, shown in percentage. The best value in each column is highlighted in blue.

3D Instance Mapping The goal of *instance mapping* is to localize object instances in 3D so a robot can distinguish objects of the same class in both image and metric space [17, 28, 44]. The main difficulty is preserving identities over long trajectories, especially when objects leave the camera's view and later reappear from different angles. Our pipeline employs SEGMAST3R to match object masks across image pairs. Given ground-truth masks and sampled pairs, we extract mask features, solve a Sinkhorn assignment, and take the row-wise argmax to obtain tentative correspondences.

Method	**IoU > 0.25**			**IoU > 0.50**			**IoU > 0.75**		
	AUPRC	R@1	R@5	AUPRC	R@1	R@5	AUPRC	R@1	R@5
SAM2 [34]	47.4	56.2	61.2	50.2	58.7	62.6	55.6	64.1	66.6
DINOv2 [30]	39.0	60.0	89.2	44.4	65.0	92.0	54.9	73.1	94.0
MASt3R (Vanilla) [22]	50.7	58.0	76.2	52.4	59.5	77.0	49.6	57.4	76.3
SegMASt3R (Ours)	84.2	91.9	99.3	87.6	94.4	99.3	89.9	94.3	96.3

Table 5: Performance on ScanNet++ [47] using noisy masks from FastSAM.

We then back-project each matched mask into 3D and drop links whose point-cloud IoU falls below 0.5, rejecting any match that fails this geometric check. Details are provided in the supplementary. Table 4 shows percentage AP versus ConceptGraphs [17]. We evaluate under two conditions: using masks generated by MobileSAM [48] as in the original ConceptGraphs setup, and using ground-truth (GT) masks for a fairer comparison of the underlying matching capability. Our geometry-aware matching yields higher accuracy in both settings, particularly when objects exit and later re-enter the field of view, demonstrating robustness to both noisy masks and challenging viewpoints.

Robustness to Noisy Segmentation Masks To assess the practical viability of our approach in real-world scenarios where ground-truth masks are unavailable, we evaluated all methods on ScanNet++ using imperfect segmentations generated by FastSAM. A key challenge in this setting is that evaluation can conflate the performance of the segment matcher with the quality of the upstream Automatic Mask Generator (AMG). A low score may not distinguish between a matching failure and an AMG failure where predicted masks do not align with any ground-truth segment.

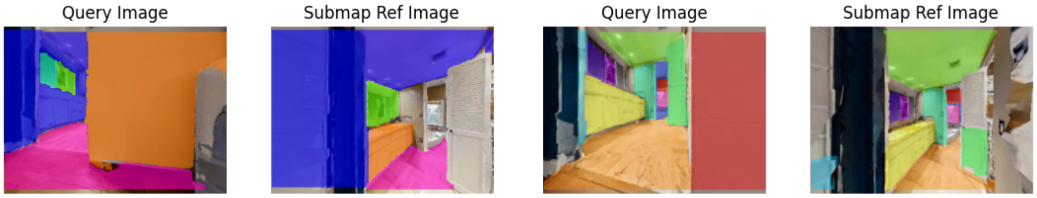
To decouple these factors and isolate the core matching performance, we adopt an evaluation protocol that does not penalize methods for AMG errors. Specifically, while all methods take noisy masks as input, the evaluation is performed only on the subset of ground-truth correspondences for which a valid match between predicted segments was possible. As shown in Table 5, SegMASt3R maintains a substantial performance margin over all baselines across different evaluation thresholds. This result confirms that the strong geometric priors learned by our method provide significant robustness, enabling superior matching even when conditioned on inconsistent and noisy segment inputs.

Object-level Topological Navigation Recent works [17, 15, 16] have explored the use of SAM2's open-set, semantically-meaningful instance segmentation for topological mapping and navigation. These methods rely on accurate segment-level association. To show the benefits of our proposed method on complex downstream tasks such as navigation, we considered an object topology-based mapping and navigation method, RoboHop [15], and swapped its SuperPoint+LightGlue based segment matching with our proposed segment matcher. This segment matching is used by RoboHop's localizer to match each of the object instances in the query image with those in the sub-map images, which is then used to estimate the currently-viewed object's path to the goal and correspondingly obtain a control signal. We used the val set of the ImageInstanceNav [21] dataset, which comprises real-world indoor scenes from HM3Dv0.2 [33]. We followed [32] for creating map trajectories and evaluating navigation performance.

In Table 6, we report navigation success for (vanilla) RoboHop and an enhanced version of it that uses SegMASt3R for segment association for localization and navigation. We use two metrics: SPL (Success weighted by Path Length) [1] and SSPL (soft SPL) [8]. Furthermore, we considered four different evaluation settings using two parameters that define the submap used for localizing the query image segments: *Submap Span* (S_s), which defines the total number of images sampled from the map based on the distance from the robot's current position, and *Submap Density* (S_ρ), which defines a subsampling factor to uniformly skip map images. These parameter configurations aid in testing narrow- and wide-baseline matching as well as the ability to avoid false positives. Table 6 shows that SegMASt3R consistently outperforms vanilla RoboHop. In particular, we achieve an absolute improvement of 27% SPL on one of the hardest parameter settings: $S_s = 16, S_\rho = 0.25$, where only 4 submap images are sampled due to a low density value. This shows that even with a very sparse submap, it is possible to maintain high navigation success rate, thus avoiding the typical trade-off between compute time and accuracy for the localizer. In Figure 2, we present a qualitative comparison between vanilla RoboHop's segment matching and that based on SegMASt3R: (left) the mismatch between the wall (orange) and the vanity cabinet for the vanilla methods leads to an incor-

Method	$S_s = 16,\ S_\rho = 0.25$		$S_s = 16,\ S_\rho = 0.5$		$S_s = 32,\ S_\rho = 0.25$		$S_s = 32,\ S_\rho = 0.5$	
	SPL	SSPL	SPL	SSPL	SPL	SSPL	SPL	SSPL
RoboHop [15]	36.34	54.25	54.51	69.98	60.57	68.29	57.52	68.47
SegMASt3R (Ours)	**63.60**	**78.84**	**63.60**	**78.33**	**66.62**	**75.20**	**63.56**	**73.89**

Table 6: Navigation performance comparison. The best value in each column is highlighted in blue.



RoboHop w. DINOv2

w. SegMASt3R

Figure 2: **Segment Matching-Guided Navigation.** (left) In vanilla RoboHop's segment matching, a wall segment (orange) gets mismatched with the vanity cabinet and misguides the agent to move towards its right, leading to a navigation failure. (right) SegMASt3R correctly recognizes the same cabinet as well as other segments and guides the robot into the bathroom, and eventually to the goal. Note that the query and submap images vary across both the methods, as we manually probed the point of failure for the baseline and the nearest agent state for ours.

rect rotation towards the right, consequently leading to navigation failure, whereas SegMASt3R's accurate segment matching (right) guides the agent into the bathroom and finally to the goal object.

Impact of the Feature Encoder To isolate the contribution of the feature encoder in our pipeline, we replace MASt3R's cross–attention block with two alternative, purely 2D backbones: *CroCo* (shared by both MASt3R and SegMASt3R) and *DINOv2*, a state-of-the-art encoder often used for off-the-shelf segment matching [17, 15, 14]. As reported in Table 7, neither backbone yields competitive segment-matching accuracy. This result underscores that learning segment descriptors from a 2D foundation model alone is inadequate; geometric context is essential. In contrast, MASt3R's cross–attention layers, 3D-aware training regimen, and explicit formulation of image matching in 3D jointly endow the model with the priors required for reliable instance association. The fact that CroCo is always second best suggests that cross-view-completion, by itself, can yield superior results for segment matching between image pairs.

Method	**0°–45°**			**45°–90°**			**90°–135°**			**135°–180°**		
	AUPRC	R@1	R@5	AUPRC	R@1	R@5	AUPRC	R@1	R@5	AUPRC	R@1	R@5
DINOv2-SegFeat	64.7	71.5	89.3	55.7	65.9	87.3	45.4	59.1	84.4	36.8	53.4	81.3
CroCo-SegFeat	73.4	78.8	92.3	64.0	73.1	90.6	50.7	64.6	87.5	38.5	56.6	84.1
SegMASt3R (Ours)	92.8	93.6	98.0	91.1	92.2	97.6	88.0	89.5	96.8	83.6	85.9	95.9

Table 7: Proposed method and model ablations performance across pose-bins on ScanNet++ [47]. Blue cells mark the best scores; Orange cells mark the second-best.

Qualitative Results Figure 4 compares SegMASt3R with SAM2's two-frame video propagation-based segment matching under *extreme viewpoint variations* on the ScanNet++ dataset. Each row shows a reference image, SAM2's matches, SegMASt3R's matches, respectively. In the top row, a wall (pink) and a door (blue) are mismatched by SAM2, whereas SegMASt3R correctly associates them, despite a very limited visual overlap. In the bottom row, SAM2 gets confused between the two monitors, whereas SegMASt3R is able to correctly associate them despite the simultaneous effect of 180° viewpoint shift and *perceptual instance aliasing* (i.e., different instances of the same object category in an image potentially lead to mismatches). Figure 3 presents qualitative results on the (outdoor) MapFree [2] dataset, where we compare our ScanNet-trained SegMASt3R with the off-the-shelf DINOv2 [30] features. Each image triplet represents the query segment (left) and its

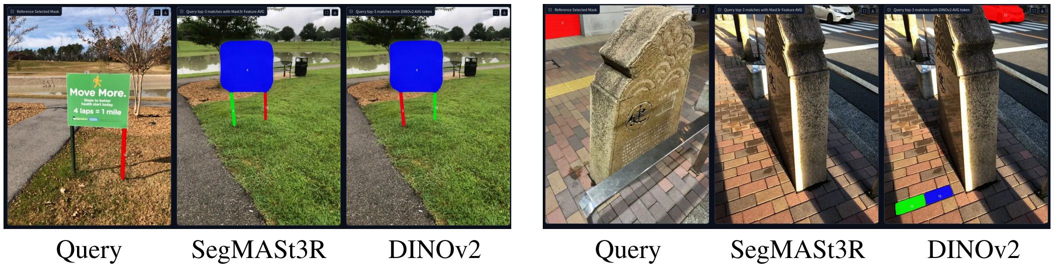


Figure 3: *MapFree Outdoor Dataset* - **Perceptual Instance Aliasing** (left): the right leg of the signboard as a query segment (red) is correctly matched by our method but mismatched with its left leg by DINOv2. **Sinkhorn Matches to Dustbin** (right): the query segment (red) is not visible in the reference image and is correctly ignored by our method, whereas DINOv2 mismatches it with a vehicle segment.

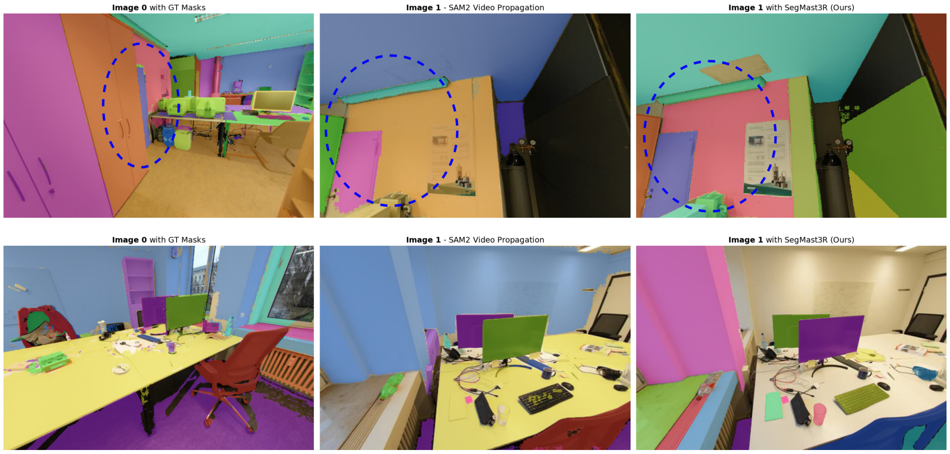


Image 0 with GT Masks
Image 1 - SAM2 Video Propagation
Image 1 with SegMast3R (Ours)
Image 0 with GT Masks
Image 1 - SAM2 Video Propagation
Image 1 with SegMast3R (Ours)

Figure 4: *ScanNet++ Dataset* – **Wide-baseline Matching** (top): The wall (pink) and the door (blue) in the query image (left) gets incorrectly associated by SAM2's video propagation (middle), whereas SegMASt3R (right) is able to correctly match them despite very limited visual overlap. **Perceptual Instance Aliasing** (bottom): unlike SAM2, SegMASt3R is able to correctly associate the pair of monitors under the simultaneous duress of an opposing viewpoint observation and perceptual instance aliasing.

retrieved matches using our method (middle) and DINOv2 (right). The query segment is displayed in red color, and its top three matches are respectively displayed in red, green, and blue colors. The left image triplet illustrates another case of perceptual instance aliasing. SegMASt3R is able to resolve this aliasing problem, whereas DINOv2 confuses the right and left legs of the signboard. The right image panel shows that our Sinkhorn solver effectively learns dustbin allocations for explicitly rejecting negatives, that is, the segments which do not have any corresponding match, whereas DINOv2 leads to incorrect matches.

6 Conclusion

We proposed SegMASt3R, a simple method to re-purpose an existing 3D foundation model MASt3R for image segment matching. Our proposed method achieves excellent results on ScanNet++ and Replica with a simple pipeline and a minimal amount of training. It especially excels on wide-baseline segment matching between image pairs. In addition, we show that SegMASt3R has practical applicability by evaluating it's performance on the downstream tasks of 3D instance mapping and object-relative topological navigation, where we significantly outperform the corresponding baselines. Overall, this work attempts to establish segment matching as a core computer vision capability, which will enable even more downstream applications in future alongside the advances in image segmentation and data association.

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[2] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Adrian Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *ECCV*, 2022.

[3] Axel Barroso-Laguna, Sowmya Munukutla, Victor Adrian Prisacariu, and Eric Brachmann. Matching 2d images in 3d: Metric relative pose from metric correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4852–4863, 2024.

[4] Gabriele Berton, Carlo Masone, Valerio Paolicelli, and Barbara Caputo. Viewpoint invariant dense matching for visual geolocalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12169–12178, 2021.

[5] Gabriele Berton, Gabriele Goletto, Gabriele Trivigno, Alex Stoken, Barbara Caputo, and Carlo Masone. Earthmatch: Iterative coregistration for fine-grained localization of astronaut photography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2024.

[6] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22877–22887, 2023.

[7] Ying Chen, Dihe Huang, Shang Xu, Jianlin Liu, and Yong Liu. Guide local feature matching by overlap estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 365–373, 2022.

[8] Samyak Datta, Oleksandr Maksymets, Judy Hoffman, Stefan Lee, Dhruv Batra, and Devi Parikh. Integrating egocentric localization for more realistic point-goal navigation agents. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 313–328. PMLR, 16–18 Nov 2021. URL `https://proceedings.mlr.press/v155/datta21a.html`.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR Deep Learning for Visual SLAM Workshop*, 2018. URL `http://arxiv.org/abs/1712.07629`.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023.

[12] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

[13] Yujie Fu, Pengju Zhang, Bingxi Liu, Zheng Rong, and Yihong Wu. Learning to reduce scale differences for large-scale invariant image matching. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1335–1348, 2022.

[14] Kartik Garg, Sai Shubodh Puligilla, Shishir Kolathaya, Madhava Krishna, and Sourav Garg. Revisit anything: Visual place recognition via image segment retrieval. In *European Conference on Computer Vision (ECCV)*, 2024.

[15] Sourav Garg, Krishan Rana, Mehdi Hosseinzadeh, Lachlan Mares, Niko Suenderhauf, Feras Dayoub, and Ian Reid. Robohop: Segment-based topological map representation for open-world visual navigation. *arXiv*, 2023.

[16] Sourav Garg, Dustin Craggs, Vineeth Bhat, Lachlan Mares, Stefan Podgorski, Madhava Krishna, Feras Dayoub, and Ian Reid. Objectreact: Learning object-relative control for visual navigation. In *Conference on Robot Learning*. PMLR, 2025.

[17] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[18] Nathan Hughes, Yun Chang, and Luca Carlone. Hydra: A real-time spatial perception system for 3d scene graph construction and optimization. *arXiv preprint arXiv:2201.13360*, 2022.

[19] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1286–1293, 2023.

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[21] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-specific image goal navigation: Training embodied agents to find object instances. *arXiv preprint arXiv:2211.15876*, 2022.

[22] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[23] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19486–19496, 2022.

[24] Siyuan Li, Martin Danelljan, Henghui Ding, Thomas E Huang, and Fisher Yu. Tracking every thing in the wild. In *European conference on computer vision*, pages 498–515. Springer, 2022.

[25] Siyuan Li, Lei Ke, Martin Danelljan, Luigi Piccinelli, Mattia Segu, Luc Van Gool, and Fisher Yu. Matching anything by segmenting anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18963–18973, 2024.

[26] Rongguang Liang, Jie Yuan, Benfa Kuang, Qiang Liu, and Zhenyu Guo. Dig-slam: an accurate rgb-d slam based on instance segmentation and geometric clustering for dynamic indoor scenes. *Measurement Science and Technology*, 35(1):015401, 2023.

[27] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023.

[28] Shigemichi Matsuzaki, Takuma Sugino, Kazuhito Tanaka, Zijun Sha, Shintaro Nakaoka, Shintaro Yoshizawa, and Kazuhiro Shintani. Clip-loc: Multi-modal landmark association for global localization in object-based maps. In *Fortieth Intl Conference on Robotics and Automation*, 2024. URL `https://arxiv.org/abs/2402.06092`.

[29] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8844–8854, 2022.

[30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[31] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 164–173, 2021.

[32] Stefan Podgorski, Sourav Garg, Mehdi Hosseinzadeh, Lachlan Mares, Feras Dayoub, and Ian Reid. Tango: Traversablility-aware navigation with local metric control for topological goals. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025.

[33] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL `https://arxiv.org/abs/2109.08238`.

[34] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[35] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[36] Xuelun Shen, Zhipeng Cai, Wei Yin, Matthias Müller, Zijun Li, Kaixuan Wang, Xiaozhi Chen, and Cheng Wang. Gim: Learning generalizable image matcher from internet videos. In *The Twelfth International Conference on Learning Representations*, 2024.

[37] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964.

[38] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[39] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[40] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[41] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European conference on computer vision*, pages 107–122. Springer, 2020.

[42] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022.

[43] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17969–17980, 2023.

[44] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5231–5237. IEEE, 2019.

[45] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In *European conference on computer vision*, pages 733–751. Springer, 2022.

[46] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15325–15336, 2023.

[47] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

[48] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

[49] Yesheng Zhang and Xu Zhao. Dmesa: Densely matching everything by segmenting anything. *arXiv preprint arXiv:2408.00279*, 2024.

[50] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We re-purpose a 3d foundation model and show how it can be used for segment matching. We show experiments showcasing the efficacy of our proposed method and down-stream applications justifying its use in real-world robotic settings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The limitations are mentioned in the supplementary.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [No]

   Justification: Paper is purely algorithmic and application oriented and does not have any theory.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: Training and dataset details are provided in Section 4.1 and Section 3.4.1 that allow for reproducibility.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release code and image pairs trained on upon paper acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies dataset and splits used in Section 4.1. Implementation details are specified in Section 3.4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: No plots requiring error bars have been presented.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Mentioned in the supplementary.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The code of ethics details are given in the supplementary.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: These are mentioned in supplementary.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Dataset details are provided in Section 4.1. Their links are given in the supplementary. The licenses can be found by following the dataset links. The backbone models used are cited throughout the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: No new assets are introduced. Only existing datasets are utilized.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We don't crowdsource data or use any human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not applicable to our submission.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMS were not used.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.