Dynam3D: Dynamic Layered 3D Tokens Empower VLM for Vision-and-Language Navigation

Bridging the gap between **Geometric Map** and **Semantic VLM** via dynamic hierarchical memory

Zihan Wang, Seungjun Lee, Gim Hee Lee

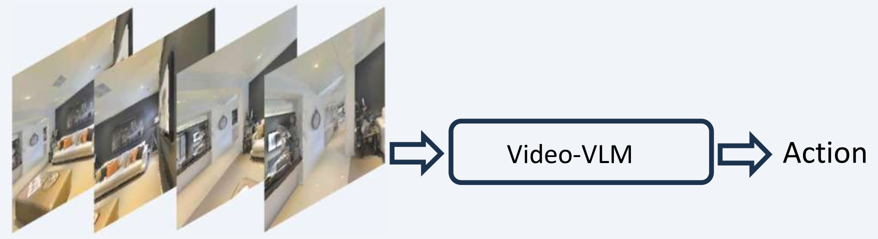
School of Computing, National University of Singapore

The Navigation Dilemma

1. The "Video Tape" Approach

Standard Video-VLMs treat the world as a linear stream of frames.

- **Spatial Amnesia:** Video-based models rely on context windows. When an object leaves the frame, it is forgotten.
- **Geometry Blindness:** 2D video frames lack explicit 3D structure, leading to collisions and poor planning.



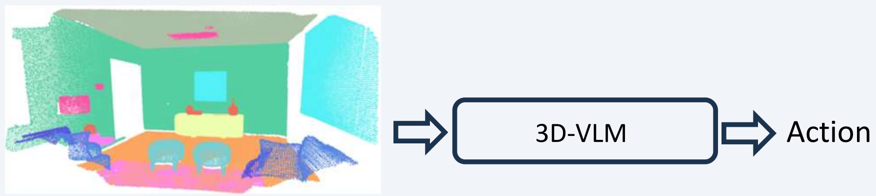
"Please go to the kitchen and take the bread out of the microwave for me."

The Navigation Dilemma

2. The "Frozen Map" Approach

Traditional 3D Maps assume a static world.

- **Static Assumptions:** Most mapping systems assume a static world, failing when object moves or environments change.
- **Granularity-Efficiency Conflict:** Dense representations (e.g., voxels) are computationally expensive for real-time reasoning, while sparse ones fail to capture fine-grained semantics for interaction.



"Please go to the kitchen and take the bread out of the microwave for me."

The Semantic Pyramid Tokenization

How do we compress a 1M-point world into a 1K-token VLM context window?

Level 1: Patch

Fine-grained Semantics

Fine-grained semantics and geometry details from CLIP features.

`Count: High`

Level 2: Instance

Object Entities

3D objects (e.g., "Chair") aggregated from patches via FastSAM masks.

`Count: Medium`

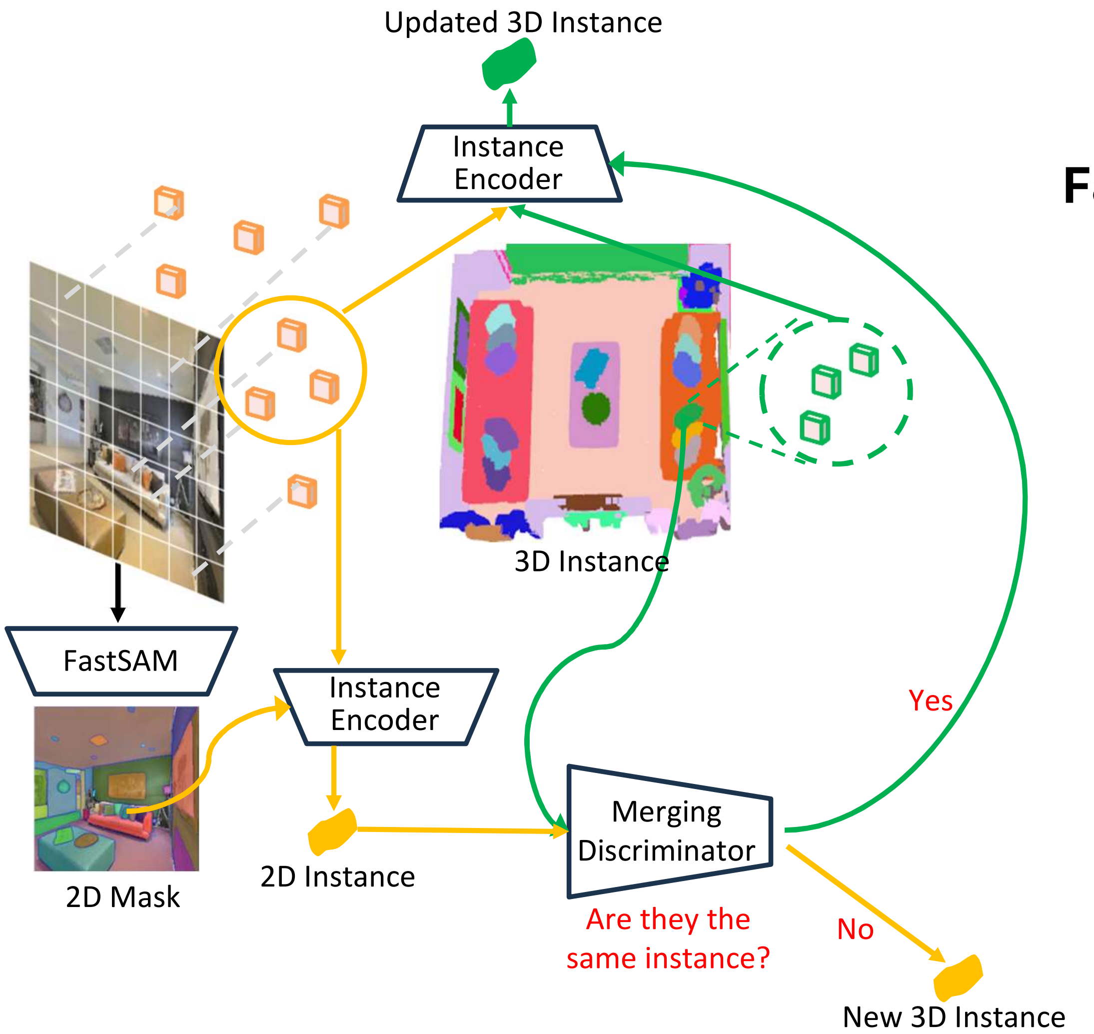
Level 3: Zone

Spatial Layout

High-level spatial regions (e.g., "Kitchen area") aggregated from objects for large-scale scene understanding.

`Count: Very Low`

Online 3D Instance Construction



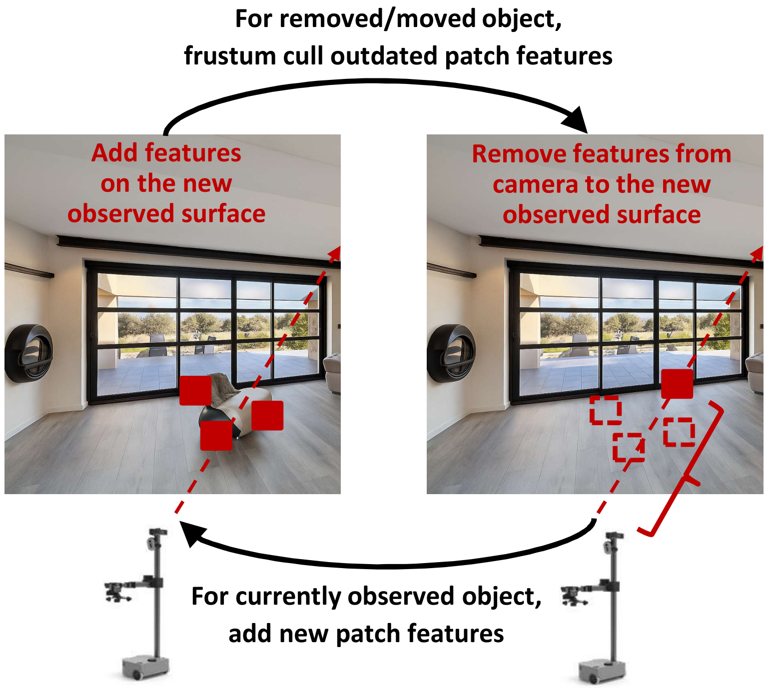
FastSAM + Merging Discriminator

1. Aggregate patches via 2D mask for 2D instance
2. Retrieve Top-K nearest existing 3D instances
3. A learned Merging Discriminator predicts if a 2D-3D pair is the same instance based on:
   - Feature Similarity (Semantic)
   - Euclidean Distance (Geometric)
4. Concatenate their patch features and update the 3D instance representation

Adapt to the Dynamic World

Video or Static maps always fail here.
We need a map that “breathes”.

Dynamic Frustum Culling (Forget outdated information)

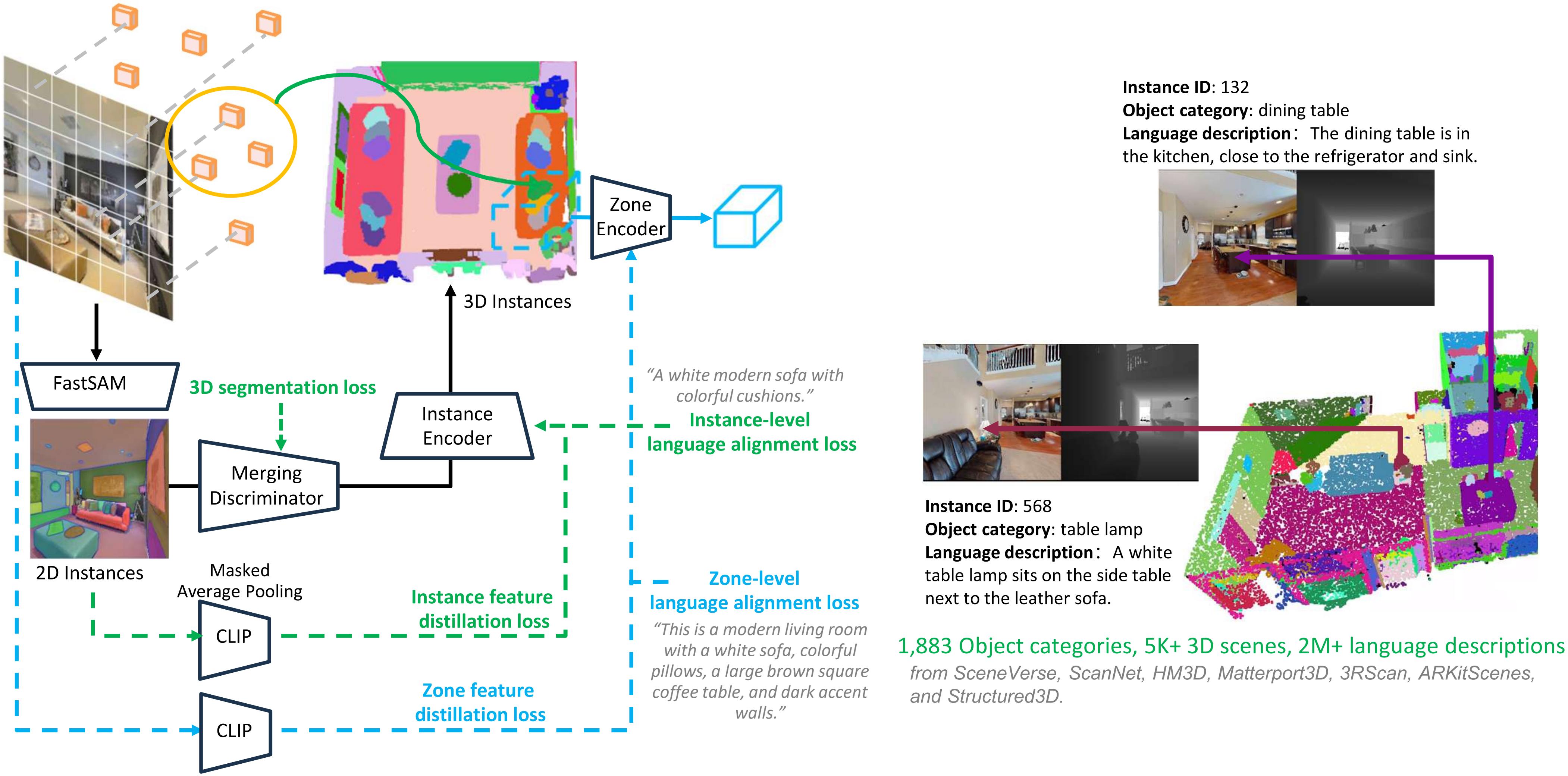


$$P_c^\top = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \mathbf{R}P_w^\top + \mathbf{T}, \quad \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z_c}\mathbf{K}\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix},$$

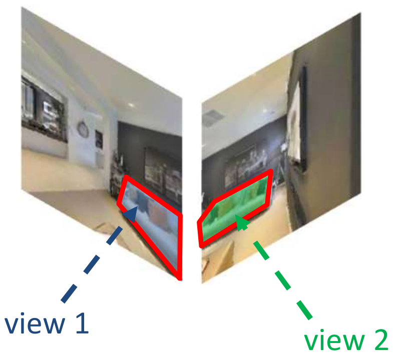
$$\text{FrustumCulling}(P_w), \text{ if } 0 < z_c < \min(d_{u,v} + \delta, \Delta), \ \ 0 < u < H, \ \text{ and } \ 0 < v < W.$$

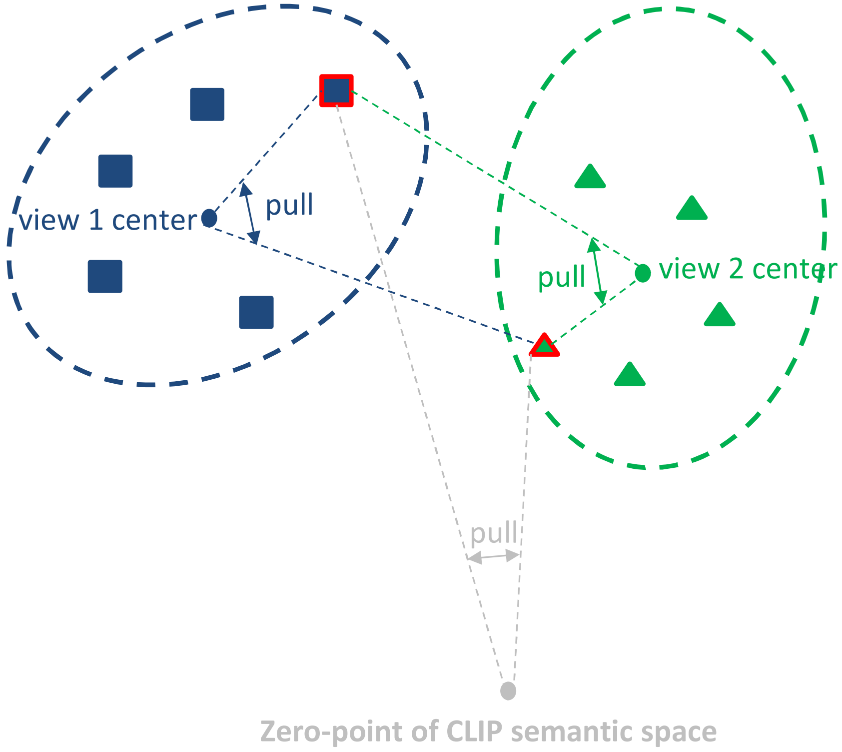
- *Where $d_{u,v}$ is the observed depth. If patch z_c is closer than the current observed surface, it will be removed.*
- *δ is a noise threshold and Δ is the farthest culling distance.*

Contrastive Learning for Semantic Alignment



Subspace Contrastive Learning for 3D Consistency





📉 The Challenge: View Inconsistency

- Naive feature distillation is interfered by **background noise**
- Results in significant feature gaps for the same instance O across different views

🛠️ The Method: Shift the Anchor

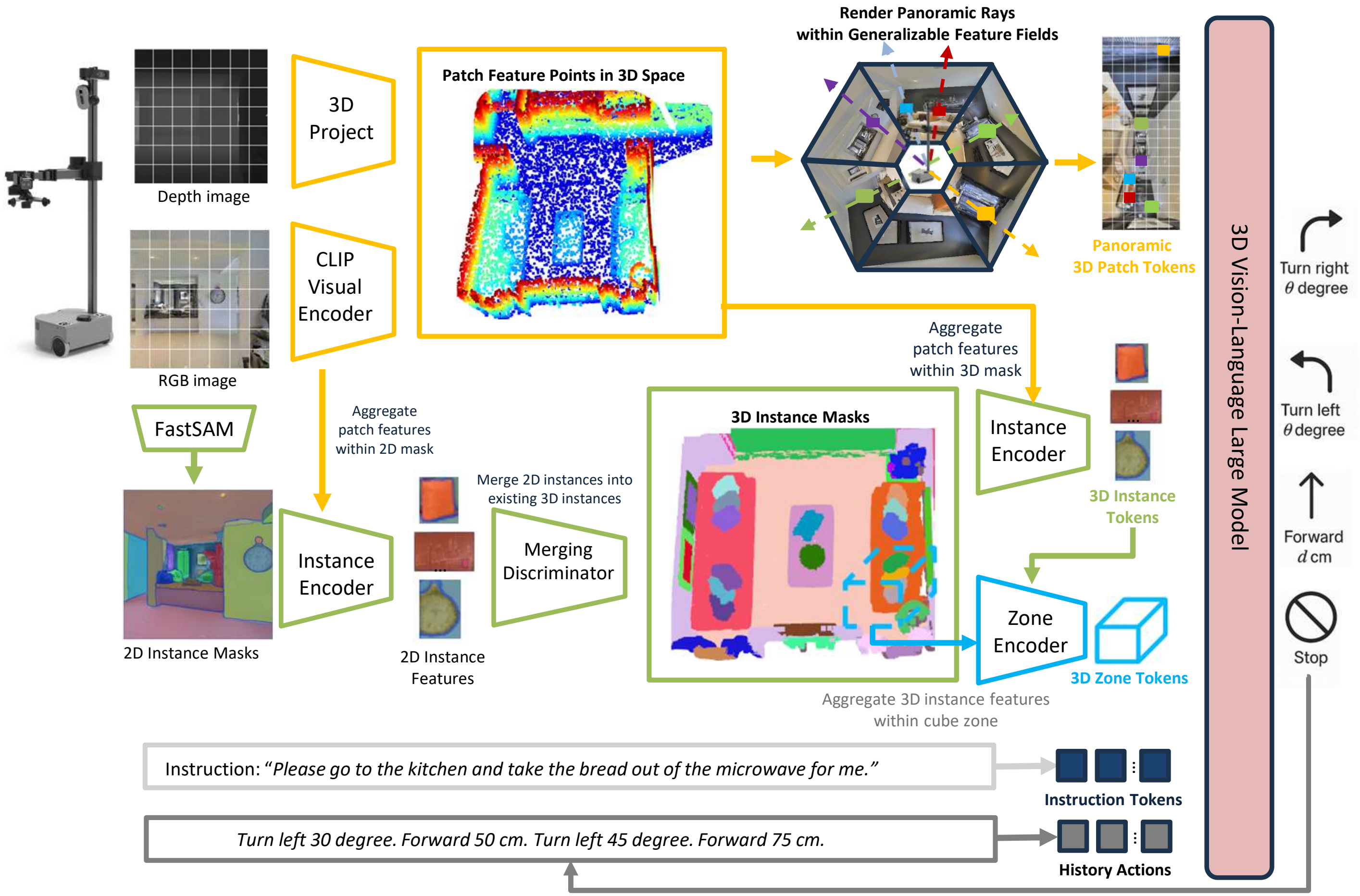
$$\mathcal{L}_{subspace\_distillation} = \frac{1}{I}\sum_{i=1}^{I} \text{CrossEntropy}(\{\text{CosSim}(\,(\mathcal{O}_i - \mathcal{V}_j), (\mathcal{O}_j^{gt} - \mathcal{V}_j)\,)/\tau\}_{j=1}^{J}, i)$$

- **Calculate V_j:** The "Semantic Center" of the current view (average of all patches)
- **Subspace Alignment:** Optimization targets $(O - V_j)$ instead of absolute O

🚀 The Effect: Bias Mitigation

- Moves the anchor from **CLIP Origin** to the **View Center** V_j
- Effectively removes view-specific bias, enforcing stronger multi-view consistency

The Brain: 3D-VLM Architecture



LLaVA-Phi-3-mini

A lightweight (3.8B) Multimodal LLM.

INPUT:

{patch_tokens}{instance_tokens}{zone_tokens}{instruction_tokens} {history_action_tokens}

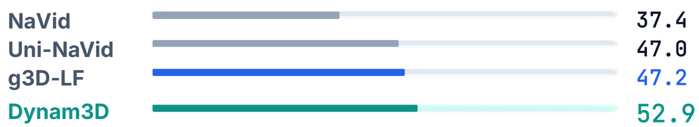
OUTPUT:

1) Turn left θ degree. 2) Turn right θ degree. 3) Forward d cm. 4) Stop.

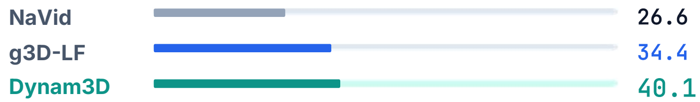
Navigation Performance

We outperform both video-based and map-based baselines, specifically in **Success Rate (SR)** and **Path Efficiency (SPL)**.

R2R-CE **Step-by-step following, *e.g.*,** *"Walk through the bedroom around the bed. Walk out of the door into the hallway. Walk towards the closet area in the hallway."*



REVERIE-CE **High-level instruction, *e.g.*,** *"Go to the familyroom and bring me the pillow from the couch closest to the entrance."*



NavRAG-CE **User-demand instruction, *e.g.*,** *"Walk to the warm hall featuring elegant wooden accents and set the large wooden table with candles and napkins for a lovely dinner ambiance."*

"After exiting the bedroom, walk straight along the hallway, then turn left at the end of the hallway to enter the kitchen, and walk to the stove."

Sim-to-Real Deployment

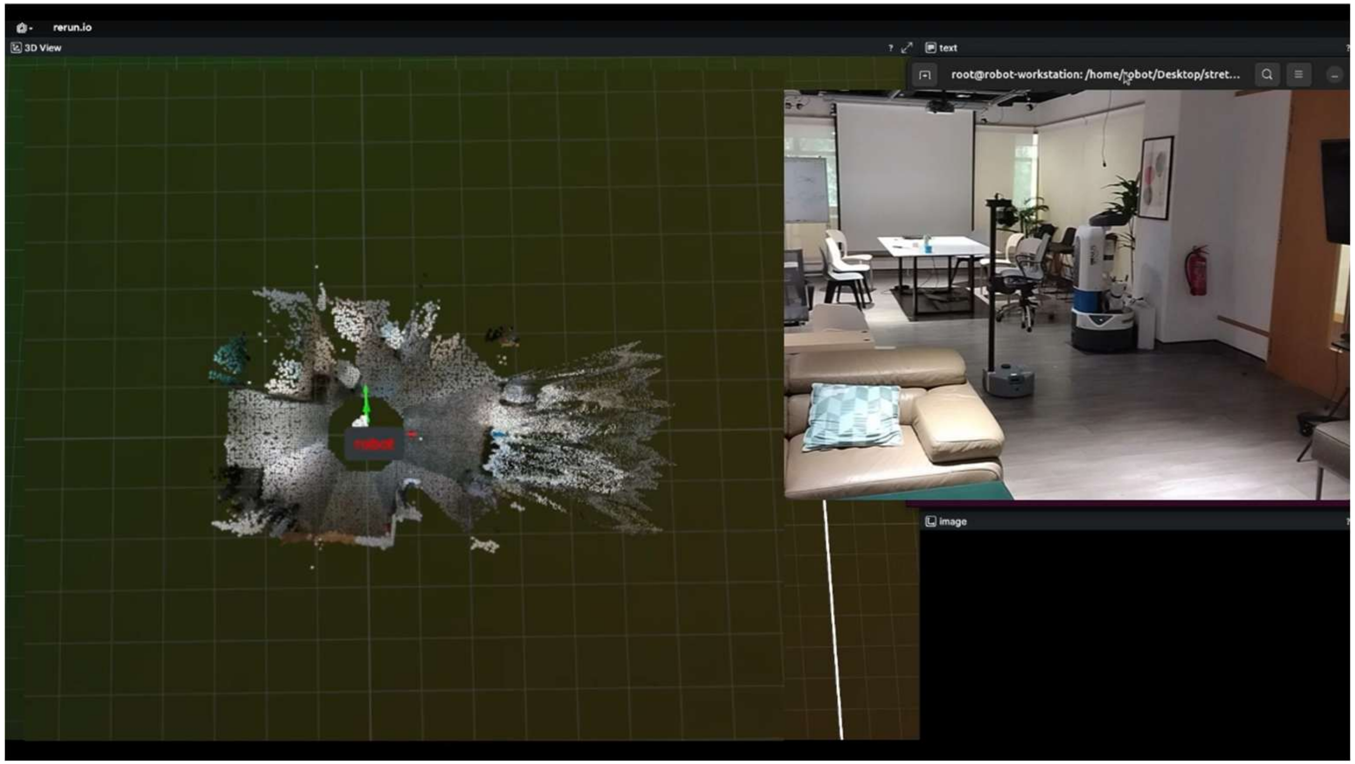
Deploy Dynam3D on **Hello Robot Stretch 3** in **NUS Robotics Living Studio**.

70% Static Success Rate

45% Dynamic Success Rate

The dynamic layered 3D tokens effectively handles moved objects.

"Please pick up the blue cup on the table and place it in the kitchen sink."



Conclusion: Towards Dynamic Embodied Memory

Hierarchical

Patch → Instance → Zone

Bridging the gap between fine-grained geometric details and high-level VLM reasoning.

Dynamic

Active Update

Frustum Culling enables the map to "breathe" and adapt to changes with **83ms** latency.

Aligned

3D Consistency

Shifting anchors to local view centers effectively denoises 2D-to-3D feature distillation.

Code Available
github.com/MrZihan/Dynam3D