OVERSEEC – Open-Vocabulary CostMap Generation from Satellite Images and Natural Language

Rwik Rana*, Jesse Quattrociocchi†, Dongmyeong Lee*, Christian Ellis†,
Amanda Adkins*, Adam Uccello†, Garrett Warnell†, Joydeep Biswas*

*The University of Texas at Austin †DEVCOM Army Research Laboratory

***Abstract*—Autonomous ground vehicles deployed in off-road settings need to reason about multiple mission-specific criteria (terrains, buildings, hazards, etc.) when planning long-range global routes. Advanced aerial imagery, captured from drones or satellite views, has the potential to provide rich prior information for such global planning. A key challenge in leveraging such aerial imagery, however, is in accommodating operator preferences *without prior knowledge of the factors, terrains, or other entities that may be needed during deployment*.**

To address these challenges, we propose OVERSEEC, a neuro-symbolic, modular, open-vocabulary, zero-shot costmap generation pipeline that produces costmaps directly from advanced aerial imagery — guided by natural-language user preferences — without requiring any domain-specific prior training. Our approach leverages: (1) a natural language-grounded semantic segmentation module (CLIPSeg [1]) to produce coarse masks from text prompts; (2) a mask refiner (SAM [2] with a lightweight rectifier) to complete and sharpen these semantic masks; and (3) a large language model (LLM) to interpret semantic entities from user preferences and generate a Python function that fuses semantic masks into a preference-aligned costmap.

We empirically demonstrate that OVERSEEC (1) produces costmaps that better reflect user preferences for global planning, achieving significantly lower rank-regret path-integral scores than state-of-the-art baselines; (2) generalizes to novel terrain classes described in natural language, even when those classes are absent from existing ontologies; (3) maintains strong segmentation accuracy and planning performance under distribution shifts compared with supervised alternatives; and (4) generates trajectories on unseen maps that human evaluators judge as closest to operator-drawn paths.

I. INTRODUCTION

Autonomous ground vehicles need to reason about multiple mission-specific criteria (terrains, buildings, hazards, etc.) when planning long-range global routes for both off-road and on-road scenarios. Aerial imagery can assist in these scenarios by providing high-quality top-down overview maps to plan global routes. While on-road navigation has seen major strides, partly due to sophisticated map software[3, 4, 5], adapting these approaches to off-road scenarios or situations requiring traversal of terrains other than roads presents significant difficulties. A core challenge arises when these systems must adapt to new ontological elements, such as previously unencountered terrain types, or to nuanced compositional user preferences that dictate complex traversal rules. Current methodologies often struggle with such adaptability, as incorporating new semantic classes or preference structures typically requires substantial modifications, including retraining segmentation models or manually re-engineering cost-assignment logic. This inflexibility is particularly acute for off-road navigation, where labeled data for all potential terrain variations is scarce, novel terrain types frequently emerge, and task-specific preferences can change rapidly. Overcoming these challenges is key to allowing costmaps to quickly adapt to new types of terrain or objects and to new user instructions without retraining it.

Current robot costmap generation typically employs a *semantics–first* approach: a fixed-ontology segmenter (e.g., U-Net [6], DeepLab [7]) assigns pixel-wise labels, followed by a manual class-to-cost mapping. This paradigm falters with (i) novel terrain classes outside its fixed ontology and (ii) complex, compositional user preferences (e.g., "prefer grass unless near buildings"), necessitating laborious retraining and rule rewriting. Representation-learning alternatives [8, 9] regress costs directly but demand task-specific data and suffer from a lack of model interpretability.

Thus, we propose a modular, zero-shot pipeline for open-vocabulary costmap generation from satellite imagery using natural language to specify preferences. Our approach integrates three specialized modules: (1) An **open-vocabulary segmentation** module (CLIPSeg [1]) generates coarse masks from arbitrary text prompts. (2) A **boundary refinement** module (SAM [2]) sharpens these masks, merging SAM's spatial precision with CLIPSeg's semantic fidelity. (3) A **preference-driven composition** module (LLM) synthesizes a Python function to combine masks according to user preferences, enabling complex logical and spatial rules. The pipeline requires no retraining, allowing instant adaptation to new classes and instructions.

II. RELATED WORK

The general problem of enabling autonomous robot navigation in complex environments requires robots to perceive their surroundings, interpret this information, and plan safe paths. Costmap generation is a critical intermediate step, translating satellite imagery and task directives into a spatial representation of traversal preference for motion planning.

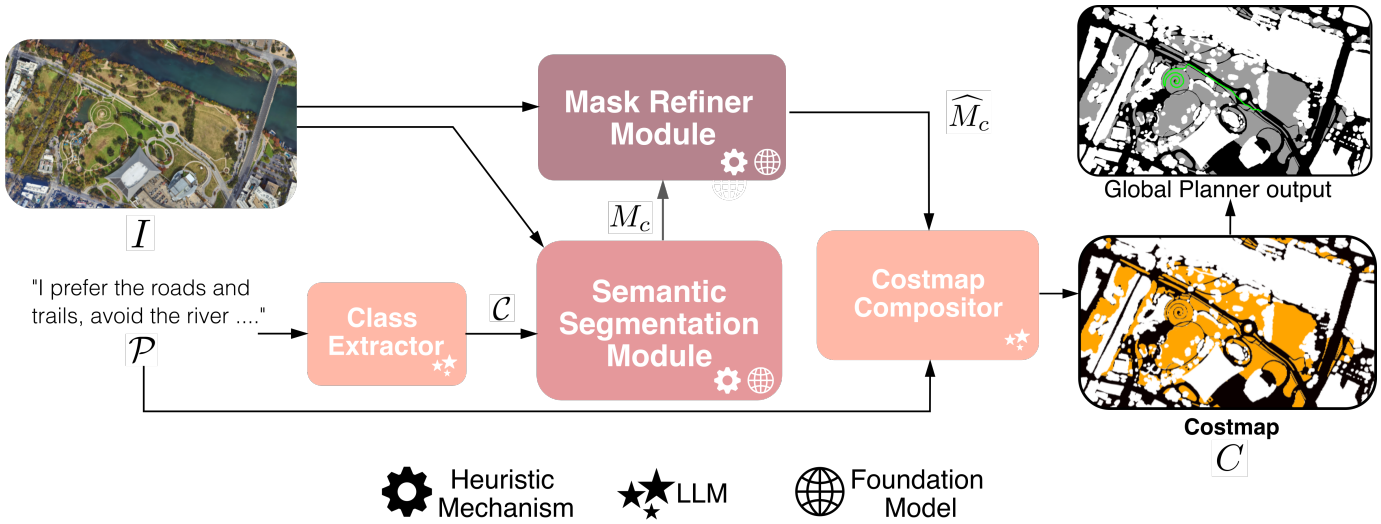

Fig. 1: **Overview of our costmap generation pipeline.** Given a satellite image I and an NL preference description $\mathcal{P}$, our system generates a preference-aligned costmap C. The *Class Extractor* (Sec. IV-A) parses $\mathcal{P}$ to identify relevant terrain classes $\mathcal{C}$. The *Semantic Segmentation Module* (Sec. IV-B) performs zero-shot segmentation over I, producing a coarse multi-class masks $\{M_c\}$, where $c \in \mathcal{C}$. The *Mask Refiner Module* (Sec.IV-C) uses $\{M_c\}$ and I to produce sharper semantic masks $\{\widehat{M}_c\}$. Finally, the *Costmap Compositor* (Sec. IV-D) uses an LLM to synthesize an executable cost assignment function from $\mathcal{P}$ and $\{\widehat{M}_c\}$, resulting in a scalar-composite costmap C used by a global planner. The top-right image shows the resulting trajectory planned over the generated costmap.

Our work addresses a specific instance: generating costmaps that dynamically adapt to novel terrain and complex user preferences. This problem lies at the intersection of remote semantic scene understanding, preference interpretation, and flexible map representations.

Broad strategies for costmap generation include: (1) The **semantics-first fixed-ontology approach** [6, 7, 10], suitable for environments with terrain classes that are assumed to be static and fully specified; however, this approach exhibits limited adaptability to novel terrain classes or dynamic user preferences. (2) **Representation Learning** [8, 9], which can learn complex functions but needs extensive labeled data and often lacks interpretability. (3) **Modular, generative, open-vocabulary systems**. Our work aligns with this third strategy, chosen for its adaptability to novel scenarios, its transparency, and its suitability for settings where labeled data is scarce and continuous retraining is impractical, such as dynamic off-road navigation. The modularity also facilitates future component-wise upgrades. For our work, we use open-vocabulary VLMs like CLIPSeg [1] for text-prompted segmentation, foundation models like SAM [2] for precise mask generation, and LLMs for code synthesis from natural language [11, 12].

Within the modular, open-vocabulary strategy, approaches like Text2Seg [13] are closely related, leveraging text-guided CLIP embeddings for remote sensing image segmentation with limited supervision. Our work builds upon such text-guided segmentation principles but differs by integrating them into a broader, fully training-free costmap generation pipeline. This system distinctively incorporates LLM-based preference composition to translate natural language instructions directly into executable costmap logic, thereby creating an end-to-end adaptable and interpretable framework.

Our work makes the following contributions:

- **Zero-Shot Test-Time Adaptability**: We introduce a pipeline that requires no training and enables adaptation to new terrain types at deployment time.
- **Human Preference Alignment**: The system generates costmaps that align with human preferences expressed in natural language.
- **Interpretable Neuro-Symbolic Approach**: We offer a human-understandable neuro-symbolic framework with tunable components.
- **Modular and Upgradable Architecture**: The pipeline is designed with modularity, allowing for the integration of new state-of-the-art foundation models or LLMs as they become available.

III. Problem Formulation

Our objective is to synthesize a scalar-valued costmap C from a satellite image I based on a user's NL preference $\mathcal{P}$, without requiring any task-specific training or extensive manual rule design. Formally, given I and $\mathcal{P}$,

$$C = f\,(I, \mathcal{P})$$

In this formulation, $I \in \mathbb{R}^{H \times W \times 3}$ represents the input high-resolution RGB satellite image of dimensions $H \times W$. $\mathcal{P}$ denotes the NL user-preference, which can describe complex ontological and compositional preferences such as "go over the trail, but avoid the puddle". The function f represents our proposed system, which takes I and $\mathcal{P}$ as inputs. The desired output is a scalar costmap $C \in [0, 1]^{H \times W}$, spatially aligned with the input image I, where lower values indicate more desirable regions for traversal.

IV. The OVerSeeC Algorithm

System Overview

We introduce OVERSEEC, a neuro-symbolic framework for open-vocabulary costmap generation from satellite imagery and natural language. The system processes two inputs: a satellite image I of the robot's operational area, and a natural language (NL) description $\mathcal{P}$, provided by an operator, detailing the navigation preferences. OVERSEEC functions in a zero-shot capacity at deployment, obviating the need for prior domain-specific training. Given I and $\mathcal{P}$ encompassing environmental details and task-specific route preferences, a four-stage pipeline is employed to generate the costmap C, as illustrated in Fig. 1:

1) **LLM-based Entity Extraction** (Sec IV-A): An LLM analyzes the NL description $\mathcal{P}$ to identify and extract a list of semantic entities (e.g., objects, terrain, etc.), necessary for subsequent reasoning about route costs, and will inform the open-vocabulary segmentation process.
2) **Open-Vocabulary Semantic Segmentation** (Sec IV-B): With I and $\mathcal{C}$ as the input, an initial coarse per-class, per-pixel segmentation mask is generated using a vision-language aligned encoder coupled with a semantic segmentation decoder.
3) **Mask Refinement** (Sec IV-C): The segmentation masks are refined in this stage. Exemplar points for each entity are extracted from the masks produced in step (2). Subsequently, a zero-shot segmentation module utilizes these exemplar points to delineate and segment all image regions corresponding to each entity.
4) **LLM-guided Costmap Composition** (Sec IV-D): The LLM, guided by the NL preference prompt $\mathcal{P}$, generates a costmap composition function. This function, expressed in a domain-specific language (DSL), operates on the entity-specific segmentation masks from step (3). It combines these masks to produce a final scalar costmap C.

A. LLM-based Entity Extraction

The user's natural-language preference $\mathcal{P}$ is parsed by an LLM (Fig. 2) to extract a set of semantic class labels $\mathcal{C}$, each assigned a category—either 'linear' (e.g., roads, trails, streams) or 'areal' (e.g., grass, trees, buildings)—and an initial processing threshold. This categorization is essential, as linear and areal entities differ in geometry and thus require distinct thresholding strategies for segmentation.

To ensure robust coverage, a fixed set of default classes $\mathcal{C}_{\text{default}}$ (e.g., 'road', 'tree', 'building', 'water', 'trail') is also included and categorized by the LLM. The final set $\mathcal{C}$ contains both extracted and default classes, each with category and threshold metadata.

$$\mathcal{C} = \texttt{LLM\_class\_extractor}(\mathcal{P}, \tau_L, \tau_A) \cup \mathcal{C}_{\text{default}}$$

This open-vocabulary extraction allows the system to operate beyond any predefined ontology and enables segmentation of novel classes (e.g., *ditches*, *orchard*) at test time.

LLM_class_extractor($\mathcal{P}, T_L, T_A$)

User Preference $\mathcal{P}$: *"I prefer the roads, the closer to the center of the road, the better. It is okay to go near the grass. But please avoid trees and buildings."*
Thresholds: $T_L = \texttt{X}$ $T_A = \texttt{Y}$
From this prompt, extract semantic classes (e.g., road, grass, tree, building, water, trails, bushes, etc.), using descriptive names if adjectives are provided (e.g., "big trees", "curved roads" from the prompt).

Do not include specific regions like *"center of road"*.

Always include "road","trail", "water", "grass", "building", and "tree" in your output by default;

For every class, determine if it is 'linear/network-like' (e.g., roads, trails, then assign T_L) or 'areal/blob-like' (e.g., grass, forest, buildings, then assign T_A).

Output **only** the dictionary mapping class names to their assigned threshold, strictly between the <DICT></DICT> markers.

```
<DICT>{"roads": X,
       "grass": Y,
       "trees": Y,
      "buildings": Y}</DICT>
```

Fig. 2: Input to the LLM combining the user preference prompt $\mathcal{P}$, thresholds T_N, T_B

B. Open-Vocabulary Semantic Segmentation

To generate terrain-specific masks aligned with $\mathcal{P}$, we first perform open-vocabulary segmentation based on classes $\mathcal{C}$ - Fig. 3. This stage identifies semantic regions without relying on fixed ontologies. The module assumes access to a *language-grounded segmentation model* (LGS)—one that accepts both an image and a natural language prompt and produces a dense per-pixel prediction for each input ontology[1, 14, 15, 16]. Enabling adaptation to arbitrary terrain categories at test time, in contrast to conventional fixed-ontology neural networks[17, 7, 18, 19].

Inputs: Satellite image I and extracted classes $\mathcal{C}$.

Per-tile inference: The high-resolution satellite image is divided into overlapping tiles $\{I^{(i)}\}$. For each tile and class $c \in \mathcal{C}$, the LGS model generates logit maps $L_c^{(i)} = g(I^{(i)}, c)$ and probability maps $P_c^{(i)} = \sigma(L_c^{(i)})$, where σ is the sigmoid function.

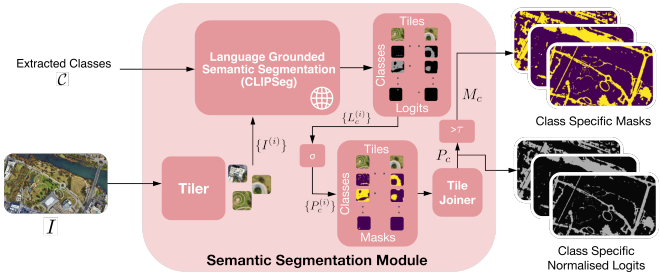

Fig. 3: Open-Vocabulary Semantic–segmentation pipeline. The image I is tiled, each tile–class pair is segmented by CLIPSeg, and tile logits are stitched and thresholded to yield per-class masks.

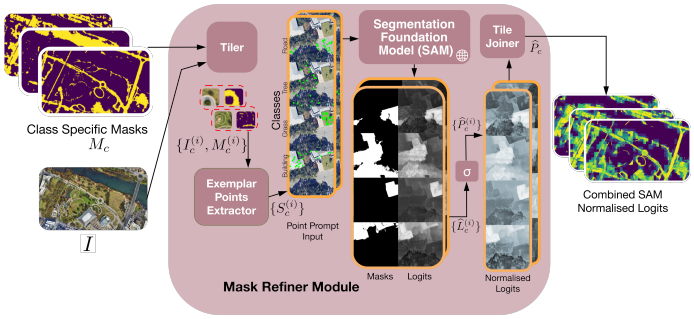

Fig. 4: Point–prompt–guided refinement. Sparse points drawn from a coarse CLIPSeg mask steer SAM toward precise object boundaries.

Stitching: Class-specific maps are reconstructed by averaging probability and logit predictions across overlapping tiles:

$$P_c(x) = \frac{\sum_i w^{(i)}(x)\, P_c^{(i)}(x)}{\sum_i w^{(i)}(x)},$$

where $w^{(i)}(x) = 1$ if $x \in I^{(i)}$.

Thresholding: Binary masks M_c are produced by thresholding the blended probabilities $P_c(x)$ using a class-dependent threshold τ_c. This threshold is set to τ_L if class c was categorized as 'linear', and τ_A if categorized as 'areal'.

$$M_c(x) = \mathbf{1}\big[P_c(x) \geq \tau_c\big].$$

C. Mask Refinement

Coarse masks $\{M_c\}$ produced by the semantic segmentation model suffer from imprecise or blurred boundaries, particularly in the presence of thin structures. To improve spatial accuracy, we employ a refinement stage using a *prompt-based segmentation model* (PSM)-that accepts sparse point prompts and returns segmentation masks [2, 18, 20, 21, 22]. Fig. 4 shows the stucture of the mask refinement module.

Inputs: Satellite image I and binary masks $\{M_c\}$.

Tiled Processing: Similar to Sec. IV-B, both the input image I and the coarse masks $\{M_c\}$ are divided into overlapping tiles. Let $I^{(i)}$ denote the i-th image tile and $M_c^{(i)}$ denote the corresponding tile of the coarse mask for class c.

Exemplar Points Generation per Tile: For each class c and for each tile $M_c^{(i)}$, we sample k_+ foreground points from pixels where $M_c^{(i)}(x) = 1$ and k_- background points from where $M_c^{(i)}(x) = 0$. This results in a point-prompt set $\mathcal{S}_c^{(i)} = \{k_+^{(i)}, k_-^{(i)}\}$ for the refinement process of that tile.

Mask Refinement on Tiles: For each image tile $I^{(i)}$ and its point-prompt set $\mathcal{S}_c^{(i)}$ for class c, the PSM, produces a refined tile-level logit $\widehat{L}_c^{(i)}$ and probability map $\widehat{P}_c^{(i)}$.

Stitching Refined Masks: Refined tile-level probability maps $\{\widehat{P}_c^{(i)}\}$ are merged into a full-sized map $\widehat{P}_c(x)$ by averaging overlaps. The final refined binary mask $\widehat{M}_c$ is obtained by thresholding $\widehat{P}_c(x)$ with a class-dependent threshold $\widehat{\tau}_c$ i.e. $\widehat{\tau}_L$ and $\widehat{\tau}_A$ for linear and areal entities respectively.

$$\widehat{M}_c(x) = \mathbf{1}\big[\widehat{P}_c(x) \geq \widehat{\tau}_c\big].$$

This refinement step enables *(i)* correction of coarse boundary artifacts and *(ii)* preservation of fine geometric details such as edges, trails, or shorelines that may be lost under morphological post-processing.

D. LLM-guided Costmap Composition

Conventional pipelines assign traversal costs with a fixed class–to–cost table, which breaks whenever preferences are *conditional* (e.g., "avoid water near roads") or reference unseen classes. We instead delegate this mapping to a large-language model (LLM) that *on the fly* synthesizes a Python function `generate_costmap(mask_dict)`. Given the refined masks $\{\widehat{M}_c\}$, the function fuses them into a single scalar costmap for the global planner.

Figure 5 shows the prompt template: it exposes three Boolean primitives—`mask_and`, `mask_or`, `mask_not`. The LLM may chain these with morphological or distance operations, providing (i) **zero-shot compositionality**, because new classes or logical rules need no retraining, and (ii) **generality**, since any spatial predicate expressible in Python can be invoked when required.

V. Implementation Details

A. Class Extraction & Costmap Code Generation

We use the instruction-tuned LLM `gemma-2-27b-it` [23] for Secs. IV-A & IV-D because it exhibits strong performance on both reasoning and code-generation tasks.

B. Open-Vocabulary Segmentation

We select **CLIPSeg** [1] as our LGS for three reasons: **(i)** its ability to handle open-vocabulary prompts, eliminating the need for retraining; **(ii)** its solid zero-shot performance on novel terrain categories, crucial for our application; and **(iii)** its architecture, which includes skip connections into the CLIP backbone, providing finer spatial detail than alternative text-guided segmenters (e.g., LSeg, MaskCLIP [15, 14]).

As described in Sec. IV-B, we employ distinct thresholds for masking. For linear features, a threshold of $\tau_L = 0.3$ is used to preserve the connectivity and prevent information loss due to weaker and intermittent activations. Areal features produce more contiguous activations, allowing for the use of a threshold of $\tau_A = 0.6$ to ensure cleaner initial masks.

Input to LLM for `generate_costmap`

```
Given a satellite image with class-wise
binary masks, generate a costmap
that reflects the user's navigation
preferences. Use the provided mask
operators (e.g., mask_and, mask_or,
mask_not) and distance-based functions
to combine these masks. The final
costmap should fully respect all
specified preferences, including
both terrain desirability and
spatial constraints. You can also
utilize functionalities from image
processing libraries such as cv2 (e.g.,
cv2.GaussianBlur, cv2.dilate)
```

```
def mask_and(mask1, mask2)-> np.ndarray:
    # Logical AND operation b/w 2 masks
    return np.logical_and(mask1, mask2).
        astype(np.uint8)
def mask_or(mask1, mask2)-> np.ndarray:
    # Logical OR operation b/w 2 masks
    return np.logical_or(mask1, mask2).
        astype(np.uint8)
def mask_not(mask)-> np.ndarray:
    # Logical NOT operation on a mask
    return np.logical_not(mask).astype(
        np.uint8)
```

```
The function you generate should have
the following format:
```

```
def generate_costmap(mask_dict):
   # mask operations
   return costmap
```

```
User Preference : "I prefer the roads,
more the center of the road the better.
It is okay to go near the grass. But
please avoid trees and buildings"
```

Fig. 5: Input to the LLM to generate a DSL(python) function for costmap generation

C. Mask Refinement

Segment Anything Model (SAM) [2] is chosen as the PSM because: **(i)** is trained on 1B masks and generalises well to aerial imagery, **(ii)** excels at recovering thin, network-like structures (*roads*, *trails*), and **(iii)** outperforms DenseCRF [20] or PointRend[18] in boundary fidelity.

To generate foreground and background exemplar points for SAM in each tile $M_c^{(i)}$ (Sec. IV-C), the number of points is set as $k_+^{(i)} = \min(100, \lceil \alpha_+ \cdot N_{fg}^{(i)} \rceil)$ and $k_-^{(i)} = \min(100, \lceil \alpha_- \cdot N_{bg}^{(i)} \rceil)$, where $N_{fg}^{(i)}$ and $N_{bg}^{(i)}$ are the number of foreground and background pixels, respectively. We set $\alpha_+ = \alpha_- = 0.005$. Capping each at 100 avoids oversampling, which can cause SAM to produce artifacts or degraded masks.

As detailed in Sec. IV-C, we use a higher threshold $\widehat{\tau}_L = 0.9$ for linear entities, as SAM can be overconfident—sometimes expanding these features unrealistically or connecting disjoint segments, thus eliminating false positives. For areal entities, we use $\widehat{\tau}_A = 0.8$ to strike a balance between preserving the full extent of the region and counteracting SAM's tendency to under-segment or erode large areas.

VI. Experiments and Results

Our evaluation is designed to answer three core research questions (RQs):

1) **Alignment and Comparative Performance**: How well do costmaps from OVERSEEC align with ground-truth semantic preferences, and how effectively do they guide planners to low-cost regions compared to state-of-the-art methods?
2) **Novel-Class Generalization**: Can the zero-shot pipeline accurately segment and assign traversal costs to terrain categories mentioned in natural-language prompts but absent from the supervised training ontology?
3) **Robustness to Distribution Shift**: How well does the system maintain its segmentation accuracy and downstream planning performance with varying geographic regions, or other visual domain shifts?

A. Experimental Setup

*1) **Baselines**:* To benchmark against conventional semantic segmentation approaches, we use 2 fixed-ontology baselines: (i) **SegFormer-B5** [17]; (ii) **DINO-UNet**, which combines a frozen ViT-DINO encoder [24] with a lightweight UNet decoder. We fine-tune SegFormer and train the DINO-UNet Decoder on a dataset $\mathcal{D}_1$ curated from OpenStreetMap (OSM) [3]. It consists of image patches of size 512×512 and in total has 6000 images.

In all the experiments henceforth, to compare against the baselines, we will replace the LGS i.e CLIPSeg with baseline fixed-ontology (FO) models i.e. SegFormer and DinoUNet, keeping all other components of OVERSEEC as is. We name them OVERSEEC`-Seg` and OVERSEEC`-Dino` respectively. We use Dijkstra's algorithm [25] to plan paths generated from these methods.

*2) **Evaluation Environments**:* To comprehensively assess OVERSEEC and baseline methods, we utilize 2 datasets :

- $\mathcal{D}_2$ to measure the performance based on ontological preferences. Ground-truth (GT) semantic maps are established by manually drawing semantic labels directly over the satellite images. This dataset consists of a combination of In-Distribution(ID), Out-of-Distribution(OOD) and Out-of-Distribution with Open-Vocabulary(OOD-OV)
- $\mathcal{D}_3$ for compositional preferences and alignment to human preferences. This dataset is more challenging and consists of OOD and OOD-OV.

Table I gives more details about the datasets.

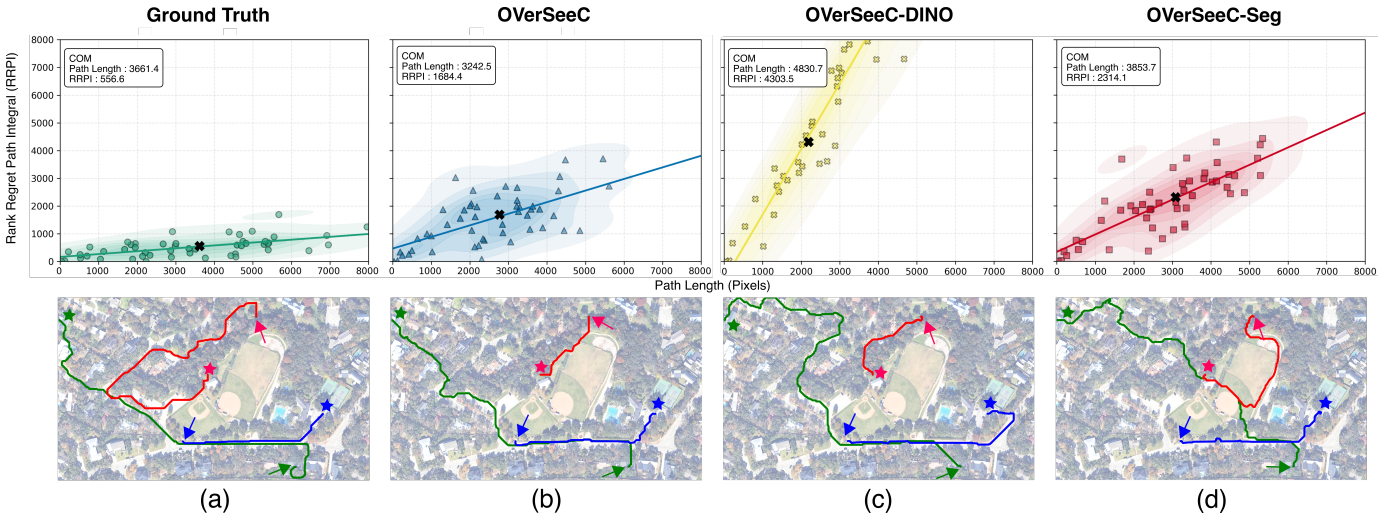

Fig. 6: **Planning results for the $\mathcal{D}_2$-OOD-OV scenario (Sec. VI).** Comparison of costmap alignment using RRPI (Sec. VI-B) metric under the user preference: *"Prefer the roads and trails, grass should be fine, try to avoid the baseball field as much as possible."* The class ranking used are: *road: 1, trail: 1, grass: 2, tree: 3, building: 4*. The top row shows RRPI vs. path length scatter plots with KDE contours; the black cross in these plots indicates the COM of the KDE, and the solid line represents a linear regression fit. A lower slope for this line is preferable, as it indicates that the RRPI score remains low even as path length increases. The bottom row shows a subset of these trajectories generated from Dijkstra's algorithm overlaid on the map (start: arrow, goal: star).

TABLE I: Evaluation settings across environment types.
ID: In-distribution (same domain as supervised training $\mathcal{D}_1$). **OOD**: Out-of-distribution. **OV**: Open-vocabulary.

Map Name	Objective
Dataset: $\mathcal{D}_2$	
$\mathcal{D}_2$-ID$_1$ $\mathcal{D}_2$-ID$_2$	ID setting with familiar regions and fixed ontology from baseline training.
$\mathcal{D}_2$-OOD	OOD region with fixed ontology.
$\mathcal{D}_2$-OOD-OV	OOD region with prompt mentioning 'baseball field' requiring OV generalization and understanding its relation to 'grass'.
Dataset: $\mathcal{D}_3$	
$\mathcal{D}_3$-HE$_1$	Recognizing novel class 'electric tower'.
$\mathcal{D}_3$-HE$_2$	Differentiating 'railway track' from roads.
$\mathcal{D}_3$-HE$_3$	OOD region with prompt mentioning 'sports fields' requiring OV generalization and understanding its relation to 'grass'.
$\mathcal{D}_3$-HE$_4$	Recognizing novel class 'river' and its relationship to 'water'.

B. Evaluation Metrics

Ranked Regret Path Integral (RRPI) Score

Quantifying the alignment of a generated costmap with user preferences is challenging, as defining an "ideal" cost function directly from NL is often intractable. However, it is generally more straightforward to establish a preference-ordered ranking of terrain types based on a given natural language description. For instance, if a user states, "trails are good, grass is okay, avoid water," we can assign ranks: trail (rank 1), grass (rank 2), water (rank 3), etc.

We introduce the **Ranked Regret Path Integral (RRPI)** score. Given a user preference $\mathcal{P}$, we first derive a rank mapping $R(c)$ for each relevant semantic class c, where $R(c) \in \{1, 2, \ldots, N_c\}$ and N_c is the number of distinct classes. A lower rank indicates a more preferred terrain type, with 1 being the lowest. The *rank regret* for traversing a pixel of class c is $R(c) - 1$. This value penalizes less preferred terrain; the most preferred has zero regret.

For any given path $\tau = [(x_1, y_1), (x_2, y_2), \ldots, (x_L, y_L)]$ of length L (i.e., the trajectory covers L pixels), through a semantic map S (where $S(x_i, y_i)$ is the semantic class of the pixel at (x_i, y_i)), the RRPI score is calculated as :

$$\text{RRPI}(\tau, S, R) = \sum_{(x,y)\in\tau} (R(S(x, y)) - 1)$$

where $R(S(x, y))$ is the rank of semantic class of the pixel (x, y). RRPI score does not inherently account for path length. To provide a more nuanced evaluation, we analyze path characteristics across two dimensions: path length (in pixels) and the RRPI score (Fig. 6). We fit a Gaussian Kernel Density Estimate (KDE) to these scatter points. The Center of Mass (COM) of this KDE blob yields an aggregate (distance, RRPI) pair that represents the performance of the method in that specific environment for the given preference prompt. For both the distance and RRPI components of the COM, lower values are considered *better*.

We calculate the RRPI scores of 50 random start and goal point pairs for each method within each map of dataset $\mathcal{D}_2$ and report the results in Table. II.

Segmentation Accuracy Metrics

We report Intersection-over-Union (IoU) on the *stitched* maps, comparing each method's per-class masks to hand-drawn ground-truth labels across all four scenarios from the RRPI evaluation.

Human Evaluation

We conducted a human study on maps from $\mathcal{D}_3$ dataset, each with multiple prompts of varying complexity. Three annotators per prompt sketched start-to-goal trajectories that best satisfied the instructions, providing behavioral references. Alignment is quantified by an *averaged* Hausdorff distance between a path τ_{sys} generated by the system and the union of the three human paths,

$$\text{HD}(\tau_{\text{sys}}) = \frac{1}{|\tau_{\text{sys}}|} \sum_{p \in \tau_{\text{sys}}} \min_{q \in \bigcup_i \tau_{h_i}} \|p - q\|_2,$$

then normalised by the map diagonal $\sqrt{H^2 + W^2}$ for cross-map comparison. *Lower* values indicate closer adherence to the human-preferred paths. We conduct our experiment on $\mathcal{D}_3$ dataset and report the results in Tables. III and IV.

.

C. Results and Discussion

RQ1: Alignment to User-Preference: OVERSEEC demonstrates alignment with user preferences across multiple evaluation modalities. Using the RRPI metric for ontological preferences (Table II), our method is competitive with supervised baselines in in-distribution (ID) settings. In the $\mathcal{D}_2$-ID$_1$ scenario, it achieves the lowest RRPI and path length among the baselines, confirming its efficacy in ID environments.

TABLE II: RRPI and path length (in pixels) of COMs across costmap generation methods and maps from $\mathcal{D}_2$. Violet highlights the best RRPI and blue highlights the best Distance among the learning-based methods for each setting.

Environment	Ground Truth		OVERSEEC-DINO		OVERSEEC-Seg		OVERSEEC	
	RRPI	Dist.	RRPI	Dist.	RRPI	Dist.	RRPI	Dist.
$\mathcal{D}_2$-ID$_1$	**889.2**	**3682**	2613.3	4749	2424.6	4837	**2379.9**	**4008**
$\mathcal{D}_2$-ID$_2$	**634.3**	**2452**	2214.2	4199	**1923.8**	**3765**	2118.2	3998
$\mathcal{D}_2$-OOD	**249.1**	**3419**	4372.4	5583	2044.5	**4085**	**1573.8**	4351
$\mathcal{D}_2$-OOD-OV	**556.6**	**3661**	4303.5	4831	2314.1	3853.7	**1684.4**	**3242**

From the human evaluation (Table III), paths generated by OVERSEEC align more closely with human-drawn trajectories (as can be seen in Fig. 7), achieving a lower normalized Hausdorff Distance. This suggests challenges for fixed-ontology models in adapting to complex, open-ontology rules specified in natural language, a task where OVERSEEC excels.

TABLE III: Normalized Hausdorff Distance (in piexls) on Human Evaluation ($\mathcal{D}_3$-HE$_x$) maps. Best results are in **bold**.

Map	OVERSEEC-DINO	OVERSEEC-Seg	OVERSEEC
$\mathcal{D}_3$-HE$_1$	0.258	0.089	**0.016**
$\mathcal{D}_3$-HE$_2$	0.027	0.073	**0.021**
$\mathcal{D}_3$-HE$_3$	0.072	0.069	**0.025**
$\mathcal{D}_3$-HE$_4$	0.090	0.051	**0.026**

Prompt-sensitivity results (Table IV) confirm that the system is parsing user intent. The table forms a confusion matrix: low diagonal Hausdorff distances mean each generated path matches its own prompt's human trajectories, whereas high off-diagonal values show that a path for one prompt fails to fit another. Thus, the LLM compositor adapts to each instruction, producing distinctly different costmaps.

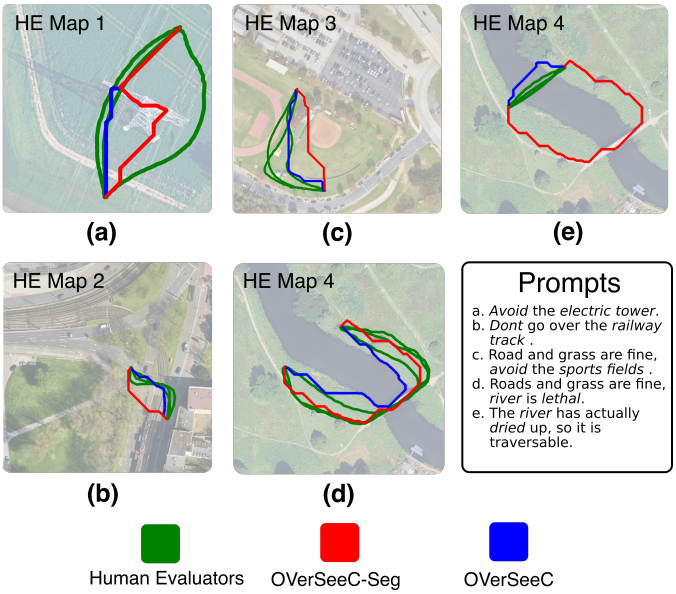

Fig. 7: **Qualitative results from human evaluation** on Dataset $\mathcal{D}_3$. Subplots (a–c) show OVERSEEC avoiding novel, open-vocabulary objects like the 'electric tower', 'railway track', 'sports fields' where the OVERSEEC-Seg fails. As discussed in Sec. VI-C:**RQ1**, OVERSEEC-Seg succeeds in (d) because of static label 'water' in its ontology. Whereas in (e) explicit understanding of river and its hierarchy with water is required and it fails to do the task. OVERSEEC succeeds in both (d) and (e).

TABLE IV: Prompt sensitivity analysis with Hausdorff Distance (in pixels). Smaller diagonal values show that the generated paths align with their corresponding prompts. For each prompt pair, the start and end goals are identical.

$\mathcal{D}_3$-HE$_1$	Human		
OVERSEEC		Prompt A	Prompt B
	Prompt A	**16.28**	172.51
	Prompt B	256.26	**54.92**

$\mathcal{D}_3$-HE$_4$	Human		
OVERSEEC		Prompt A	Prompt B
	Prompt A	**163.78**	892.97
	Prompt B	351.13	**151.92**

The river cases in Fig. 7(d–e) highlight the benefit of contextual understanding. With the prompt "river is lethal" (d), both OVERSEEC and OVERSEEC-Seg avoid the river, the latter relying on its static *water* class. When the prompt changes to "the river has dried up, so it is traversable" (e), only OVERSEEC updates the costmap accordingly, whereas OVERSEEC-Seg, still treating *water* as lethal, takes a far less efficient route.

Figure 8 shows that the costmap compositor (Sec. IV-D) handles geometric prompts: from a single image (a) it produces costmaps that drive the planner along the road's *center* (b) or its *sides* (c), assigning low cost to roads while distinguishing centerline from edges.

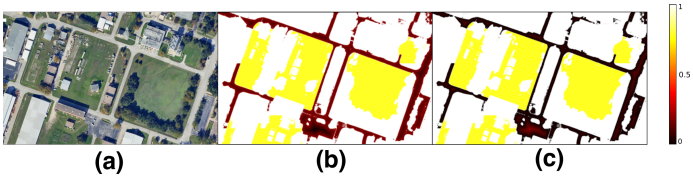

Fig. 8: Qualitative examples of intricate compositional preference alignment. (a) Input satellite image. Costmaps generated by OVERSEEC for the preference (b) "stay in the middle of the road" and (c) "stay on the side of the road".

RQ2: Novel-Class Generalization: The capabilities of OVERSEEC become more apparent when dealing with novel classes. In the OOD-OV scenarios (Table II and III and Fig. 7), which introduces an unseen novel classes like 'baseball field', 'sports fields' 'river', 'electric tower', 'railway track', OVERSEEC achieves a lower RRPI score and significantly lower Hausdorff distance than both OVERSEEC-Seg and OVERSEEC-DINO. The trajectories in Fig. 6 illustrate this difference: OVERSEEC-Seg based paths incorrectly traverse the baseball field, whereas OVERSEEC navigates around it. We see a similar trend for all cases as shown in Fig. 7. This shows OVERSEEC's ability to generalize to unseen classes and complex instructions simultaneously, a key aspect of its zero-shot, open-vocabulary design.

TABLE V: Class-based IoU for overall semantic segmentation quality on stitched maps (hand-drawn labels). Method$^-$ denotes pipeline without SAM refinement

Method	Tree	Grass	Building	Water	Road/Trails
OVERSEEC-DINO$^-$	0.298	0.084	0.224	0.335	0.281
OVERSEEC-DINO	0.346	0.105	0.212	0.210	0.359
OVERSEEC-Seg$^-$	0.410	0.544	0.398	**0.802**	0.403
OVERSEEC-Seg	0.392	0.289	0.350	0.435	0.482
OVERSEEC$^-$	**0.682**	**0.581**	0.640	0.697	0.543
OVERSEEC	0.623	0.517	**0.644**	0.665	**0.569**

RQ3: Robustness to Distribution Shift: OVERSEEC demonstrates robustness to distribution shifts, both in preference alignment and underlying segmentation quality. In the OOD setting (Table II), our method achieves a lower RRPI score than the supervised baselines, indicating better adherence to preferences when encountering novel geographies.

From a segmentation perspective, (Table V), OVERSEEC outperforms the supervised methods, showcasing effective generalization. The SAM-based refiner consistently improves performance on linear entities like roads and trails. For areal classes, a slight trade-off between boundary precision and minor erosion is observed, but overall performance remains strong. This robust segmentation, combined with effective preference alignment in diverse environments, suggests OVERSEEC is well-suited for real-world deployment.

VII. Limitations and Future Work

While OVERSEEC demonstrates promising results in zero-shot, preference-aligned costmap generation, several aspects present opportunities for further refinement and investigation.

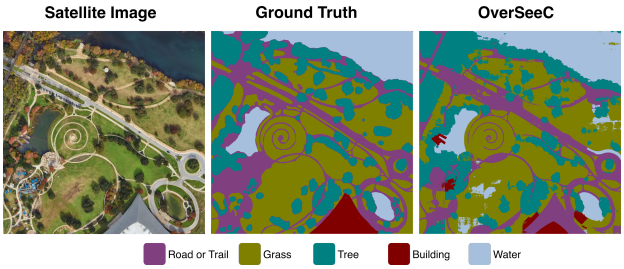

Fig. 9: Qualitative comparison of semantic segmentation for the ID-1 scenario. From left to right: input satellite image, ground truth segmentation, and segmentation output from OVERSEEC.

1) **Tight Integration**: The exemplar points for guiding SAM may not always be optimal for mask refinement. Exploring nuanced methods like SAMRefiner[21], or alternatively, exploring shared embeddings or joint optimization might improve both efficiency and consistency.
2) **Handling of Nuanced Semantic Relationships**: The system's current approach to complex semantic hierarchies relies on the LLM's interpretation during cost composition, which might not always capture the full nuance. More explicit hierarchical semantic reasoning, possibly via graph-based methods [26] could enable a more robust understanding of inter-related, and multi-label semantic classes.
3) **Robustness to Visual Artifacts and Occlusions**: Perception modules can be sensitive to visual artifacts like sharp shadows, which can degrade segmentation accuracy. Additionally, the system does not currently model occlusions. Enhancing robustness to such artifacts and developing occlusion reasoning capabilities, perhaps using contextual cues or generative inpainting, could lead to more coherent costmaps and improved planning.

VIII. Conclusion

We presented OVERSEEC, a novel neuro-symbolic, modular, and zero-shot architecture for generating costmaps from aerial imagery using natural language preferences, addressing the critical need for adaptability in off-road navigation without requiring fine-tuning. By leveraging language-grounded segmentation, mask refinement, and LLM-driven preference interpretation, OVERSEEC enables rapid adaptation to new classes and compositional instructions. Empirical evaluations demonstrated its high adaptability, successful generalization to novel terrains and preferences, and superior performance over traditional baselines in challenging out-of-distribution and open-vocabulary scenarios. This work highlights the potential of combining large-scale pre-trained models in neuro-symbolic frameworks for creating adaptable, user-centric robotic navigation systems.

Acknowledgments

This work is partially supported by the ARL SARA (W911NF-24-2-0025). Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, June 2022.

[2] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[3] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

[4] Google LLC. Google Maps. https://maps.google.com, 2025.

[5] Apple Inc. Apple Maps. https://maps.apple.com, 2025.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[8] Luisa Mao, Garrett Warnell, Peter Stone, and Joydeep Biswas. pacer: Preference-conditioned all-terrain costmap generation. *IEEE Robotics and Automation Letters*, 10(5):4572–4579, 2025.

[9] Kavan Singh Sikand, Sadegh Rabiee, Adam Uccello, Xuesu Xiao, Garrett Warnell, and Joydeep Biswas. Visual representation learning for preference-aware path planning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 11303–11309, 2022.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500, 2023.

[12] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023.

[13] Jielu Zhang, Zhongliang Zhou, Gengchen Mai, Mengxuan Hu, Zihan Guan, Sheng Li, and Lan Mu. Text2seg: Remote sensing image semantic segmentation via text-guided visual foundation models, 2024.

[14] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision (ECCV)*, 2022.

[15] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022.

[16] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. Generalized decoding for pixel, image, and language, 2022.

[17] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[18] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9796–9805, 2020.

[19] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[20] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[21] Yuqi Lin, Hengjia Li, Wenqi Shao, Zheng Yang, Jun Zhao, Xiaofei He, Ping Luo, and Kaipeng Zhang. Samrefiner: Taming segment anything model for universal mask refinement. *ArXiv*, abs/2502.06756, 2025.

[22] Lemeng Wu Xiaoyu Xiang Fanyi Xiao Chenchen Zhu Xiaoliang Dai Dilin Wang Fei Sun Forrest Iandola Raghuraman Krishnamoorthi Vikas Chandra Yunyang Xiong, Bala Varadarajan. Efficientsam: Leveraged masked image pretraining for efficient segment anything. *arXiv:2312.00863*, 2023.

[23] Gemma Team and Co. Authors. Gemma: Open models based on gemini research and technology, 2024.

[24] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[25] Edsger W Dijkstra. A note on two problems in connexion

with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[26] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. Deep hierarchical semantic segmentation. *arXiv preprint arXiv:2203.14335*, 2022.