2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# OmniVLA: An Omni-Modal Vision-Language-Action Model for Robot Navigation

Noriaki Hirose[1,2], Catherine Glossop[1], Dhruv Shah[3] and Sergey Levine[1]

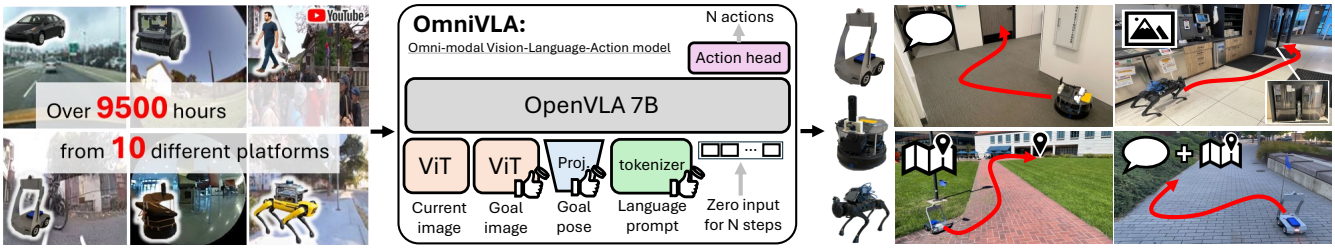



**Fig. 1:** We train a highly generalizable vision-based navigation policy with flexible conditioning, leveraging over 9,500 hours of data collected across 10 different platforms. Our policy supports diverse goal modalities, including language prompts, goal poses, goal images, and their combinations, and can control a variety of robot platforms.

***Abstract*— Humans can flexibly interpret and compose different goal specifications, such as language instructions, spatial coordinates, or visual references, when navigating to a destination. In contrast, most existing robotic navigation policies are trained on a single modality, limiting their adaptability to real-world scenarios where different forms of goal specification are natural and complementary. In this work, we present a training framework for robotic foundation models that enables omni-modal goal conditioning for vision-based navigation. Our approach leverages a high-capacity vision-language-action (VLA) backbone and trains with three primary goal modalities: 2D poses, egocentric images, and natural language, as well as their combinations, through a randomized modality fusion strategy. This design not only expands the pool of usable datasets but also encourages the policy to develop richer geometric, semantic, and visual representations. The resulting model, OmniVLA, achieves strong generalization to unseen environments, robustness to scarce modalities, and the ability to follow novel natural language instructions. We demonstrate that OmniVLA outperforms specialist baselines across modalities and offers a flexible foundation for fine-tuning to new modalities and tasks. We believe OmniVLA provides a step toward broadly generalizable and flexible navigation policies, and a scalable path for building omni-modal robotic foundation models.**

## I. Introduction

When navigating in an environment, humans naturally use various modalities of information to infer and discover paths to goals (e.g., looking up a GPS location in a map, visual landmarks, and following language directions). However, prior work in robot navigation typically trains policies with single modalities based on narrow applications. When goals are nearby, it is convenient to describe them in a language (for example, "move along the building and go to the entrance", while further goals can be described more effectively as GPS coordinates. However, a truly generalist navigation policy must be able to perform tasks that require us to leverage multiple sources of information to be successful. For example, it might be most relevant to specify a combination of GPS coordinates and landmark images for tasks like autonomous delivery or inspection. While these modalities overlap significantly, they provide complementary information about the task and how it should be performed, particularly in the context of the partial view of the world that the robot receives from its sensors. Motivated by this need, this study aims to train a generalizable policy capable of navigating with goal specifications expressed in multiple modalities. By training on omni-modal goals, we aim to enable stronger and more flexible policies, ultimately acquiring a foundation model that exhibits high adaptability to novel modalities and unseen environments.

Training foundation models requires that we leverage as much data as possible. With recent data collection efforts in the robotics community, it has become feasible to train powerful generalist navigation policies. However, these policies are typically trained with only a single kind of task representation, such as egocentric images, 2D poses, or natural language. This limits the datasets that can be used to those that accord with the desired task representation (e.g., only datasets with language labels), restricts how the model can be used at test time, and potentially limits how the model processes the task and observation – i.e., more spatial or visual tasks might improve the model's spatial reasoning abilities, while language tasks might provide a complementary benefit in enhancing understanding of semantics, much like task mixtures for multi-modal language models [1, 2].

In this study, we propose a family of **Omni**-Modal **V**ision-**L**anguage-**A**ction Models (**OmniVLA**) for autonomous navigation that can ingest goals expressed in multiple modalities, leveraging information across modalities, and achieving a more flexible navigation policy. We train our model with goals specified through three primary modalities: (1) 2D poses, (2) egocentric images, and (3) natural language. By simultaneously learning to interpret these different modalities, the model must develop a richer understanding of the geometric, visual, and semantic information of the task, resulting in a more powerful navigation model. Moreover, our method allows the user to instruct the robot with multiple modalities, making it more user friendly and directly allowing the policy to leverage more than one kind of information about a goal. For example, a user can specify a target pose and provide instructions on *how* to reach it through language.

[1]UC Berkeley, [2]Toyota Motor North America, [3]Princeton University

To train these policies, we compose several design choices into one system, resulting in a flexible and general navigation policy. We use an expressive vision-language-action (VLA) model as the base model [3], enabling us to leverage internet-scale knowledge from the VLM backbone [4] and the representations learned during fine-tuning on cross-embodiment robot data [5]. As a result, our policy exhibits strong generalization and fine-tuning capabilities, following language instructions not seen in the training data, and adapting to completely new modalities. Additionally, we address the problem of modality imbalance and scarcity by using modality dropout during training, and modality masking during inference. This ensures that our policy attends to all available goal modalities and learn from cross-modal goal representations across all datasets.

OmniVLA is the first end-to-end VLA model for navigation that unifies diverse task modalities. Our results demonstrate strong performance across all modalities, surpassing *specialist* baselines trained for a single task or goal modality. Furthermore, we demonstrate that OmniVLA can be efficiently fine-tuned to new goal modalities and new environments using a limited dataset. We believe OmniVLA will serve as a valuable resource for future navigation research, both as a recipe for a generalist model and as a pre-trained checkpoint for fine-tuning to specific modalities.

## II. Related Work

In navigation tasks, we can specify goals with various modalities, such as egocentric images [6–9], 2D poses [10, 11], and language [12–15]. For egocentric image-conditioned navigation, prior works [6–9] combine publicly available datasets across various robot embodiments to train generalist policies. These policies work well in indoor environments, which require rich visual information and do not allow for reliable access to GPS. 2D pose-conditioned policies are most successful in long-horizon tasks in outdoor environments with GPS localization [10]. MBRA[11] introduces a model-based reannotation approach to leverage large-scale data sources to perform more challenging long-distance navigation tasks conditioned on 2D goal poses.

Language-conditioned navigation offers a user-friendly, flexible interface to instruct robots to reach a specific target object or area in a given environment, even over long distances [12, 16, 17]. However, robust instruction-following demands language understanding beyond simple object references. Early works relied on pre-trained language encoders [12, 18], while recent approaches use powerful VLM backbones, either using them directly to perform navigation tasks [19] or fine-tuning them on robot data [20]. Other works have extended to using counterfactual action generation [14] and non-robot data [15] to improve training of these models. LeLaN [13] leverages both robot and non-robot data to learn a generalized language-conditioned navigation policy, using a model-based approach to generate counterfactual actions that reach target objects along with language prompts derived from VLM reasoning. While such synthetic action commands and language prompts enable scalable training [13–15], their inaccuracy can become a performance bottleneck.

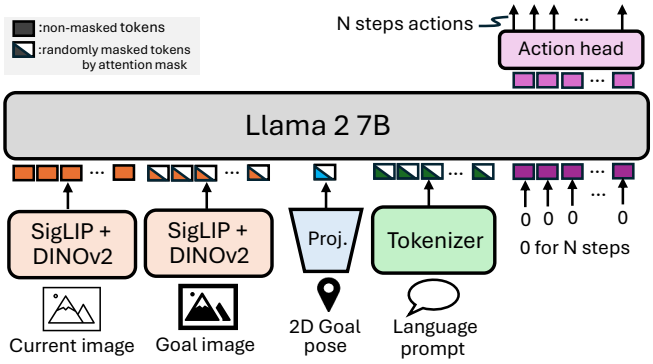



**Fig. 2: Network architectures for multi-modal vision-based navigation.** Our design builds on existing large VLA checkpoints, adding a visual backbone and a projector to condition on egocentric goal images and 2D goal poses. During training, we randomly mask tokens for these modalities and the language prompt.

In manipulation, recent robotic foundation models (RFMs) aim to unify vision, language, and action for generalization [21, 22]. Evolving from early multi-modal transformers to large-scale systems focused on dexterity and real-world deployment, they mark a shift toward scalable frameworks capable of robust, versatile robotic control [5, 23, 24]. Some manipulation works leverage large datasets with different observation and action spaces as well as conditioning (language, poses, etc.) by masking unavailable inputs during training [23, 25, 26] and demonstrate the benefits of simultaneously training on multiple input types [27].

Motivated by the success of previous work in the manipulation domain, OmniVLA goes beyond prior navigation work to leverage as much robot and non-robot data as possible across multiple modalities and embodiments, learning a policy that can condition on goal images, language, and GPS. Notably, OmniVLA is trained on almost 10,000 hours of real-world navigation data, which is the largest pre-training dataset for an end-to-end navigation policy to our knowledge.

## III. An Omni-Modal VLA for Navigation

We present OmniVLA, an end-to-end navigation policy with omni-modal conditioning. OmniVLA effectively captures core navigation capabilities, such as collision avoidance and path following, and can perform complex goal-conditioned navigation behaviors through omni-modal user inputs. Furthermore, to train OmniVLA, we leverage numerous public datasets—each containing at least one modality—resulting in a model with strong generalization and improved adaptability to new modalities and environments.

### A. OmniVLA Architecture

We build OmniVLA on top of a high-capacity VLA architecture, which contains knowledge from Internet-scale pre-training (for the base VLM) and cross-embodiment robot actions (for the base VLA). We modify the architecture to

| Dataset | | | | Action | | | Modality | | | | Environment |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Platform | Max. speed | Used Hrs | Raw | Hz | Labels | Lang. | Pose | Ego. | Sate. | |
| GNM mixture [6] | 6 platforms | 0.5 – 5.0 m/s | 62.0 | ✓ | 3.0 | raw | | ✓ | ✓ | | Off-road, office, sidewalks |
| LeLaN mixture [13] | 3 platforms | 0.5 – 1.0 m/s | 128.7 | | 3.0 | NoMaD | ✓⋆ | ✓⋆ | | | Office, home, sidewalks |
| Frodobots-2K [28] | ERZ | 1.0 m/s | 700 | ✓ | 3.0 | MBRA | | ✓ | ✓ | ✓ | Sidewalks |
| BDD-V [29] | Car | 20.0 m/s | 8680 | ✓ | 1.0 | MBRA ⋄ | | ✓ | ✓ | | Road |

**TABLE I: Training dataset mixture.** Our training mixture consists of 9,500 hours across 10 different platforms, including human-collected data, and covers a diverse set of environments. GNM and LeLaN are themselves mixtures of 7 and 5 (respectively) publicly available datasets. The details are shown in the Appendix D. We use the satellite modality in the Frodobots dataset for evaluation of OmniVLA's ability to adapt to a new task. ⋆ indicates labels that are synthetically generated, and ⋄ indicates that a customized version of MBRA was used to generate the refined action labels, described in the Appendix B.

enable flexible goal-conditioning and omni-modal representation learning, while preserving the strong vision-language priors in the base model.

Fig. 2 illustrates the network architecture, built on top of OpenVLA [3], a 7B-parameter VLA model. We process the robot's current visual observations using a visual encoder. For goal-conditioning, our architecture supports three different modalities — visual, positional, and language — which can also be specified simultaneously. We project each individual goal modality into a shared token space, which serves as the input of the LLM backbone. We use "modality dropout" to flexibly mask the goal modalities during training and inference. For the action output, we follow OpenVLA-OFT [30] and add a linear action head to the LLM output to generate a sequence of actions $\{\hat{a}_i\}_{i=1...N}$.

We also implement a smaller "edge" version of our OmniVLA architecture built on top of ViNT [8], a 50M-parameter navigation transformer. We modify this to add multiple goal encoders and perform modality dropout in the same way as our base model (see Appendix A). In real-world evaluation IV, we find OmniVLA-edge to be a very compelling choice for resource-constrained deployment, where inference of large VLA models is not feasible.

### B. Training OmniVLA

While using multi-modal inputs is enticing, training policies to accept omni-modal inputs requires compiling robot datasets that support training and addressing the relative imbalance and scarcity of the available modalities.

**Training data.** Table I lists the publicly available datasets for vision-based navigation. This data is comprised of 13 publicly available datasets and contains 9,500 hours across 10 different embodiments. While most of these datasets include manually collected labels, we also include data that has synthetic labels, which may be noisier, and actions refined with a re-annotation process. To our knowledge, we compile the largest mixture of navigation datasets with highly diverse environments, embodiments, and modalities to train OmniVLA.

While large datasets enable generalization, large-scale data collection efforts can result in more noise and therefore, be less accurate. Following [11] and [13], we use synthetic actions generated with MBRA [11] for the Frodobots dataset and NoMaD [9] for the LeLaN dataset during training. Since existing reannotation approaches cannot account for the large embodiment gap of the BDD-V [29] dataset (an autonomous vehicle dataset vs. the small robot datasets we use otherwise), we train a reannotation model to generate reasonable synthetic actions, making it possible to use this data for training, similar to [11]. However, we found it was reasonable to train directly with the high-quality raw actions available in the GNM mixture. The details about the BDD-V [29] dataset are shown in the Appendix B.

**Training method.** Following prior work for manipulation VLAs [23], we use a form of random dropout to train on all available modalities, resulting in a more efficient and generalized model. To train OmniVLA, we construct a batch of samples from the datasets included in Table I. For each sample, we then independently sample from the available goal modalities to form the conditioning input, which we call $t_m$. For example, for the GNM mixture, the conditioning input can be chosen from 2D pose, egocentric goal image, or their combination. Naturally, we get coverage over all modalities and datasets while using this dropout mechanism to improve training stability.

For each training step, we construct an attention mask that excludes unused or unavailable modalities (filled with random values) so that only the selected modalities are attended to. Training on these mixed-modality batches encourages the model to better represent goal information, yielding improved representations for generalization and fine-tuning.

We train the OmniVLA policy described by $\{\hat{a}_i\}_{i=1...N} = \pi_\theta(I_c, I_g, p_g, l_g, t_m)$, where $I_c$, $I_g$, $p_g$, $l_g$ and $t_m$ are the egocentric current image, the egocentric goal image, the 2D goal pose, the language prompt and the randomly selected modality, respectively. We calculate the objective $J$ with $J_{il} = \frac{1}{N}\sum_{i=1}^{N}(a_i^{ref} - \hat{a}_i)^2$ to imitate the $N$-step action reference $\{a_i^{ref}\}_{i=1...N}$ and update $\theta$ to minimize $J$. Following [13], the examples in the LeLaN dataset use an additional task-specific objective to encourage object reaching behavior, where the final action is trained to be close to the target object. Details of the objectives are shown in the Appendix E.

**Training details.** We set the action chunk size $N$ to 8 at 3 Hz, corresponding to 2.4 seconds for all models. For each batch, we sample the data at a ratio of LeLaN : GNM : Frodobots : BDD-V = 4:1:1:1 to balance modalities. Since we cannot secure a sufficiently large batch size for some models even on a server with multiple GPUs, we accumulate the gradient for several steps to stabilize the training process. In training OmniVLA with OpenVLA checkpoints on eight H100 GPUs, we use a per-GPU batch size of 7 and accumulate gradients for 4 steps, yielding an effective batch size of 224 (=7×8×4). In addition, we apply LoRA

to limit the learnable parameters to about 5 %, allowing us to maximize the batch size and balance training speed with training stability. Note that LoRA is only used for the OpenVLA-based model due to its large model size. The other training settings, such as learning rate, language tokenization, normalization, and so on, are the same as the default setting in the original code for each model type.

## IV. Experimental Setup

We begin by describing our setup for evaluating omni-modal navigation on our real-world robot platforms. We conduct extensive real-world evaluations and compare against state-of-the-art specialist and generalist baselines.

### A. Navigation tasks

We consider the following three navigation tasks:

**Language-conditioned navigation:** We evaluated OmniVLA on language prompts that not only direct the robot toward the target location but also specify how it should behave along the way. We conducted evaluations in 40 environments, including an office, a kitchen area, an entrance hall, a public park, and sidewalks, using diverse language prompts. The goals were placed 5–30 meters from the robot's initial position. To assess the benefit of large pre-trained models, we introduced out-of-distribution (OOD) language prompts that go beyond the instructions present in the training data. While the training data primarily contained object-reaching instructions such as 'move toward X', we designed OOD prompts that specified how to navigate toward the target in half of the trials. Following CAST [14], we crafted these prompts and evaluated the robot's behavior based on its adherence to instructions. We also selected 17 environments where obstacles were placed between the robot's start and the target, making the tasks more challenging and testing the core navigation abilities of the policy.

**Egocentric goal image-conditioned navigation:** With egocentric goal images, our policy is tasked with navigating the robot to target locations up to 3 meters away. To extend this range, we follow prior vision-based approaches [8, 31, 32] and employ topological memory for navigation in 8 different environments, enabling navigation to more distant goals. To build the goal graph, we record image observations at 1 Hz. During deployment, we initialize from the first observation and, at each time step, estimate the closest node as the current location, as in [8, 9]. The image from the next node is then provided as the goal image $I_g$ to our policy.

**Goal pose-conditioned navigation:** When conditioned on 2D goal poses, our policy navigates to targets 25–100 meters from the initial robot position. We use GPS to estimate the robot's location and the location of the goal. At each step, we compute the pose of the robot relative to the target pose, $p_g$. For evaluation, we selected 7 environments and ran 3 trials at different times in each, accounting for GPS jitter.

### B. Mobile Robot Platforms

We evaluate OmniVLA on the FrodoBots ERZ platform [33], a low-cost mobile robot. It is equipped with multiple sensors, including front and rear cameras, GPS, an IMU unit (gyroscope, accelerometer, and compass), and wheel velocity sensors on all four wheels. All sensor streams are accessible through a web API, and our trained navigation policies interface with the robot by sending linear and angular velocity commands.

To assess cross-embodiment generalization, we evaluate our policies on two additional platforms: the VizBot [34] wheeled robot and the Unitree Go1 [35] quadruped.

### C. Baselines

In our evaluation, we compare OmniVLA against seven baselines across all modalities. This includes model-free navigation policies trained from scratch [8, 9, 11], methods that can leverage internet-scale videos [13], methods using off-the-shelf visual representations like CLIP [12], as well as state-of-the-art VLA models [14]. For LeLaN, CounterfactualVLA, ViNT, MBRA-pose, and MBRA-image, we use the authors' original implementation and checkpoints for evaluation. Below, we describe two baselines that differ slightly from their original implementations.

**NoMaD [9]:** For 2D goal pose-conditioned navigation, we run the NoMaD policy in exploration mode to generate 30 candidate trajectories. We then select the trajectory whose final predicted position is closest to the target pose $p_g$ and use it to control the robot. For egocentric goal image-conditioned navigation, we follow the original NoMaD implementation using the goal image $I_g$.

**CoW [8]:** For language-conditioned navigation, we provide the current observation and prompts describing the target object to the OWL-ViT B/32 detector [36], reported as the strongest model in the original paper, to estimate the object's bounding box. Following [13], we crop the point cloud within the box and compute the median point as the target object pose. To ensure fair comparison with our approach, which relies solely on a single RGB camera without depth or LiDAR, we estimate depth using Depth360 [37] and project it to reconstruct the point cloud. A state lattice motion planner is then used to generate velocity commands.

**Other VLA backbones:** To further understand the role of VLA architectures and pre-training, we also implement our omni-modal goal-conditioning strategy for the 1B MiniVLA [38] and the 500M SmolVLA [39]. Please see the Appendix C for details on these architectures.

## V. Evaluating Omni-Modal Navigation

To evaluate our OmniVLA policies, we focus our experiments on answering the following questions:

- **Q1** Does omni-modal pre-training outperform single-modality navigation policies?
- **Q2** Can OmniVLA follow a composition of multiple goal modalities?
- **Q3** Can OmniVLA be adapted to new goal modalities, environments, and embodiments?

### A. OmniVLA vs. single-modality policies

For **Q1**, we conduct a comparative analysis with single-modality policy baselines, with results summarized in Table II. The evaluation metrics are provided in the table

| Method | Language | | | | 2D Pose | | Image | |
|---|---|---|---|---|---|---|---|---|
| | SR | Behavior | $SR^S$ | $SR^C$ | SR | Prog. | SR | Prog. |
| CoW [12] | 0.30 | 0.05 | 0.55 | 0.24 | – | – | – | – |
| LeLaN [13] | 0.43 | 0.15 | 0.64 | 0.35 | – | – | – | – |
| CounterfactualVLA [14] | 0.33 | 0.45 | 0.18 | 0.24 | – | – | – | – |
| MBRA-pose [11] | – | – | – | – | 0.86 | 0.92 | – | – |
| NoMaD [9] | – | – | – | – | 0.33 | 0.47 | 0.63 | 0.77 |
| ViNT [8] | – | – | – | – | – | – | 0.50 | 0.68 |
| MBRA-image [11] | – | – | – | – | – | – | **1.00** | **1.00** |
| SmolVLA [39] | 0.10 | 0.15 | 0.00 | 0.12 | 0.38 | 0.70 | 0.38 | 0.45 |
| MiniVLA [38] | 0.23 | 0.15 | 0.27 | 0.12 | 0.43 | 0.75 | 0.25 | 0.36 |
| OmniVLA-edge | 0.60 | 0.25 | 0.82 | 0.47 | 0.91 | 0.95 | **1.00** | **1.00** |
| OmniVLA | **0.73** | **0.65** | **1.00** | **0.65** | **0.95** | **0.98** | **1.00** | **1.00** |

**TABLE II: Quantitative analysis for conditioning on single modalities.** SR and Prog. indicate the success rate and the partial progress towards the goal, respectively. "$SR^S$" averages over simple experiments without obstacles. "$SR^C$" averages over complex experiments with obstacles in the environment. Behavior indicates the success rate in following OOD language prompts.

| Train Modality | Dataset | | | | Test Modality | | | |
|---|---|---|---|---|---|---|---|---|
| | GNM | LeLaN | Frod. | BDD | Lang. | 2D pose | Ego. image | Sate. image |
| Language | | ✓ | | | 0.43 | – | – | – |
| 2D pose | ✓ | | ✓ | | – | 0.86 | – | – |
| Ego. image | ✓ | ✓ | ✓ | | – | – | **1.00** | – |
| Sat. image | | ✓ | | | – | – | – | 0.19 |
| Omni-modal | ✓ | ✓ | ✓ | ✓ | **0.60** | **0.91** | **1.00** | **0.57** |

**TABLE III: Ablation study of multi-modal training OmniVLA-edge.** We evaluate the performance of OmniVLA-edge when trained on single and multiple modalities and evaluated with single modality task representations.

caption. Our largest model, OmniVLA, demonstrates a clear advantage over each modality-specific baseline. This improvement stems from our policy's ability to learn more generalized navigation capabilities from our large, highly diverse training mixture, which is substantially larger than the datasets available for individual modalities. Additionally, we observe that the choice of pre-trained VLA architecture and pre-training data has a significant impact on the performance. Notably, OmniVLA demonstrates strong generalization on language- and pose-conditioned tasks, outperforming the strongest single-modality specialist.

**Architecture comparison.** While all the OmniVLA variants demonstrate the ability to generalize across modalities, OmniVLA outperforms the smaller VLA models by a wide margin. We notice that MiniVLA and SmolVLA perform quite poorly, which can be primarily attributed to the wide gap between their pre-training domains (MiniVLA primarily trained on manipulation data) and limited capacity (SmolVLA might not have enough capacity to learn useful cross-embodiment representations). In contrast, OmniVLA-edge adopts a more specialized architecture for navigation tasks (described in the Appendix A), which enables it to achieve exceptional performance for its size. However, there remains a substantial gap between OmniVLA-edge and OmniVLA on language following, highlighting the benefits of the vision-language priors inherited from pre-trained VLMs.

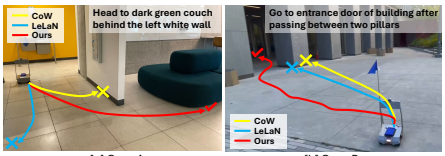
**Image-conditioned performance.** OmniVLA matches the performance as the strongest single modality baseline, MBRA, achieving 100% success, and outperforming NoMaD and ViNT. While NoMaD and ViNT are trained on the more limited GNM mixture, MBRA leverages a much more diverse pre-training dataset that enables strong generalist performance. OmniVLA is trained on a combination of these data sources, and is able to match this strong performance.

**Pose-conditioned performance.** OmniVLA achieves a 9% performance improvement in success rate (binary) and partial progress towards the goal over the strongest *specialist* baseline, MBRA-pose. While both policies leverage the same large datasets for pose-conditioned navigation, with a high-capacity model and additional data, OmniVLA is able to outperform prior methods. This observation is particularly impressive for VLA-based policies, as this is not well represented in the VLM/VLA pre-training.

**Language-conditioned performance.** OmniVLA demonstrates a large improvement in goal-reaching and language following on the diverse set of language prompts we evaluate. While both OmniVLA and LeLaN are trained only with in-domain language prompts from the LeLaN dataset, the performance gap between OmniVLA and LeLaN is particularly evident on "Behavior", which measures success in following OOD language prompts. This result suggests that leveraging a larger pre-trained LLM provides stronger priors for navigation tasks. OmniVLA also outperforms CounterfactualVLA, even though CounterfactualVLA is trained with more diverse language prompts, as OmniVLA leverages a larger backbone and a larger training mixture, which makes it more general and flexible to OOD prompts. We also evaluated NaVILA [15] using our prompts in the same environment. However, NaVILA fails, scoring 0.0 on all metrics, due to a domain gap in prompt style: it requires detailed, step-by-step instructions such as "Turn right, move straight, and stop when you see X," even when X is already visible and nearby, which is not in line with the open-set, natural language prompt format we target.

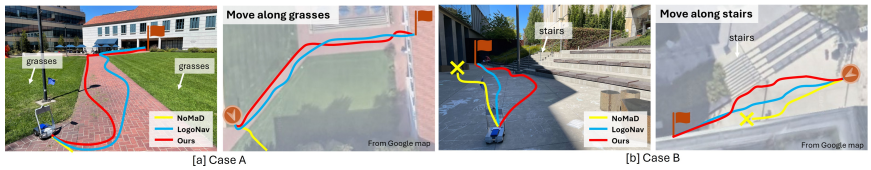
Figure 3 illustrates rollouts of OmniVLA and the baselines on two language-conditioned navigation tasks. Both scenes are particularly challenging as the target objects are not visible from the robot's initial position, requiring the policy to use the other information in the prompt to reach the goal. Our method successfully follows the language instructions and reaches the target object, whereas the baselines LeLaN and CoW fail, navigating instead toward the incorrect object.

**Dataset ablation.** We conduct an ablation study to highlight the benefits of training OmniVLA with a larger and more diverse data mixture while keeping the model architecture fixed. We train versions of OmniVLA-edge, each on a single modality, and compare them with our multi-modal OmniVLA-edge policy. Table III summarizes the datasets used for training model and reports the goal-reaching rates for each task. For this evaluation, we also introduce satellite images as a goal modality and assess performance under the same environments and setup as 2D goal-pose navigation. As shown in Table III, OmniVLA demonstrates clear advantages, particularly in language- and satellite image-conditioned navigation. By jointly training across multiple modalities, OmniVLA learns core navigation behaviors from all datasets and achieves stronger generalization to unseen environments compared to single-modality policies trained on limited data. In addition, we observe that including highly




[a] Case A [b] Case B

**Fig. 3: Visualization of language-conditioned navigation rollouts.** OmniVLA can follow OOD language instructions in various indoor and outdoor environments.




[a] Case A [b] Case B

**Fig. 4: Visualization of goal pose- and language-conditioned navigation rollouts.** Conditioned on OOD language and a goal pose, our policy can perform complex, long-horizon navigation tasks, reaching distant goals (⚑) while following behavioral instructions.

| Method | SR | Prog. | Behavior |
|---|---|---|---|
| SmolVLA [39] | 0.10 | 0.35 | 0.20 |
| MiniVLA [38] | 0.30 | 0.53 | 0.00 |
| NoMaD [9] | 0.40 | 0.57 | – |
| MBRA-pose [11] | 0.70 | 0.80 | – |
| OmniVLA [30] | **0.80** | **0.86** | **0.60** |
| OmniVLA-edge [8] | 0.30 | 0.56 | 0.10 |

**TABLE IV:** OmniVLA excels at omni-modal navigation, reaching a goal position (*where*) while following language instructions (*how*).

| Modality | Fine-tune | SR | Prog. |
|---|---|---|---|
| Sat. goal image | | 0.57 | 0.75 |
| Sat. goal image | ✓ | 0.83 | 0.92 |
| 2D goal pose | | 0.81 | 0.90 |
| 2D goal pose | ✓ | 0.86 | 0.96 |

**TABLE V: Adapting OmniVLA - edge to new environments.** We adapt our model using only 1.2 hours of data, focusing on satellite goal images and 2D goal poses.

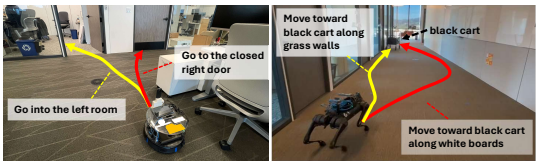



**Fig. 5: Deploying OmniVLA on multiple embodiments.** We deploy our policy on the Vizbot and Unitree Go1 robots. Our policy can follow natural language instructions out of the box and reach the targets.

varied cross-embodiment data, the BDD-V dataset, enables the policy to succeed in roughly half of the failed cases of a policy trained without BDD-V, highlighting the ability of OmniVLA to ingest and benefit from diverse data sources.

### *B. Omni-modal conditioning with OmniVLA*

By training on omni-modal task representations, OmniVLA can learn to follow multiple goal signals. Towards answering **Q2**, we conduct experiments where tasks are specified by providing both 2D goal poses (*where?*) and behavioral language instructions (*how?*) in 10 different environments. In each environment, we provide a 2D position located 25–100 meters from the robot's initial position, along with a language prompt that specifies a behavior that the robot must follow to reach the goal, such as "move along the wall", "move on the grass", and "move between objects A and B". These prompts are out-of-distribution (OOD) and not included in the training dataset. To our knowledge, no existing work has explored such a challenging composition of modalities in navigation.

Since no existing methods can handle both modalities simultaneously, we compare our approach to *specialist* baselines trained with 2D pose conditioning (NoMaD and MBRA-pose, Table IV). OmniVLA demonstrates the ability to attend to the information of both the prompt and the goal pose, achieving 80% success and experiencing only a 5% drop in language capabilities from Table II. The smaller OmniVLA variant fails to handle the language instructions due to limited modal capacity. The other VLA baselines (SmolVLA and MiniVLA) are also unable to solve the task.

### *C. Adapting OmniVLA to new goal modalities*

To evaluate OmniVLA's capabilities as a "foundation model" for navigation and answer **Q3**, we assess three aspects: (i) learning a new goal modality, (ii) fine-tuning on new datasets, and (iii) controlling new robot embodiments. For (i), we set up a controlled experiment by pre-training OmniVLA-edge *without* the satellite modality. We then freeze most of the model and replace the vision encoders for egocentric image goals with one for satellite images, training only the swapped encoder. With this model, we evaluate the adaptability of our model to leverage new goal modalities. We find that OmniVLA-edge can indeed adapt to this new modality, improving significantly over a specialist policy trained only for satellite goal navigation (0.19 $\rightarrow$ 0.62). This suggests that the cross-modal representations learned by OmniVLA can facilitate learning new tasks, while retaining useful navigation affordances learned during pre-training.

To evaluate (ii), we conduct two experiments to investigate OmniVLA's capability to adapt to new data: (a) adapting to new environments using a small dataset, and (b) learning a new language domain from a new available dataset. For (a), we collect 1.2 hours of data, including 2D goal poses and satellite images, from test environments unseen during training, and fine-tune the trained OmniVLA-edge to evaluate adaptability. Table V shows that the fine-tuned policy improves performance on both modalities, suggesting that OmniVLA can effectively adapt to new environments. For (b), we fine-tune our OmniVLA using the recently published CounterfactualVLA dataset, which contains more diverse language prompts (see [14]). To stabilize fine-tuning, we sample batches to maintain the balance between modalities used in our prior training. Specifically, half of the data previously sampled from LeLaN is replaced with CounterfactualVLA data when fine-tuning OmniVLA. The fine-tuned policy achieves scores (SR, Behavior, $SR^S$, $SR^C$) = (0.825, 0.700, 1.000, 0.765), improving over the original scores (0.725, 0.650, 1.000, 0.647), demonstrating the ability of OmniVLA to learn new language domains from emerging datasets. Because the CounterfactualVLA dataset provides both raw and counterfactual robot actions corresponding to diverse language prompts, our policy learns better collision avoidance and instruction-following, with notable improvements in "Behavior" and "$SR^C$".

To show the generality of our model trained on data from several robots for (iii), we demonstrate deployment on

two additional platforms: the wheeled mobile robot VizBot and the quadruped robot Go1. We mount different cameras on the robots and test them on the most challenging language-conditioned navigation tasks. As shown in Fig. 5, our best OmniVLA enables both robots to follow language instructions and reach the target locations. Detailed robotic behaviors are provided in the supplemental videos.

## VI. CONCLUSIONS

In this work, we introduced OmniVLA, an omni-modal vision-language-action model for robot navigation. Our policy flexibly interprets multiple goal modalities, including language prompts, 2D poses, egocentric goal images, and their combinations, within a unified framework. By initializing from pre-trained VLAs, OmniVLA leverages Internet-scale knowledge while being trained on over 9,500 hours of robotic navigation experience across ten robot platforms.

Extensive real-world evaluations show that OmniVLA consistently outperforms prior baselines, achieving stronger performance across modalities, robust out-of-distribution generalization, and the ability to follow diverse multi-modal tasks. Furthermore, we demonstrate foundation model qualities, including adaptability to new modalities (e.g., satellite imagery), efficient fine-tuning with extra data, and cross-embodiment transfer across different robot platforms.

We believe OmniVLA represents a significant step toward broadly generalizable navigation policies. To further advance this direction, we will release our models and training code as a resource for developing scalable and flexible robot navigation models. In particular, there remains room for improvement in language-conditioned navigation, both in goal-reaching and instruction-following. Progress in this area will benefit greatly from larger and more carefully curated datasets, which we hope the community will pursue.

## ACKNOWLEDGMENTS

This research was supported by Berkeley AI Research at the University of California, Berkeley and Toyota Motor North America. And, this work was partially support by ARL DCIST CRA W911NF-17-2-0181 and the NSF under IIS-2150826. We thank Kyle Stachowicz and Satoshi Koide for discussing the model architecture and its implementation to train our OmniVLA. And we thank Frodobots AI for providing robot hardware for our evaluations.

## REFERENCES

[1] J. Lin *et al.*, "Vila: On pre-training for visual language models," in *Proceedings of the CVPR*, 2024, pp. 26 689–26 699.
[2] M. Deitke *et al.*, "Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models," in *Proceedings of the CVPR*, 2025, pp. 91–104.
[3] M. Kim *et al.*, "Openvla: An open-source vision-language-action model," *arXiv:2406.09246*, 2024.
[4] S. Karamcheti, K. McKinzie, T. Huang, S. Gong, K. Fang, K. Lin, A. Li, B. Lee, F. Ma, A. Mandlekar, *et al.*, "Prismatic VLMs: Investigating the design space of visually-conditioned language models," in *International Conference on Machine Learning*, vol. 235. PMLR, 2024, pp. 23 123–23 144.
[5] A. O'Neill *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *Proceedings of the ICRA*. IEEE, 2024, pp. 6892–6903.
[6] D. Shah *et al.*, "Gnm: A general navigation model to drive any robot," in *Proceedings of the ICRA*. IEEE, 2023, pp. 7226–7233.
[7] N. Hirose *et al.*, "Exaug: Robot-conditioned navigation policies via geometric experience augmentation," *arXiv:2210.07450*, 2022.
[8] D. Shah *et al.*, "Vint: A foundation model for visual navigation," *arXiv:2306.14846*, 2023.
[9] A. Sridhar *et al.*, "Nomad: Goal masked diffusion policies for navigation and exploration," in *Proceedings of the ICRA*, 2024, pp. 63–70.
[10] D. Shah and S. Levine, "Viking: Vision-based kilometer-scale navigation with geographic hints," *arXiv:2202.11271*, 2022.
[11] N. Hirose *et al.*, "Learning to drive anywhere with model-based reannotation," *arXiv:2505.05592*, 2025.
[12] S. Y. Gadre *et al.*, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proceedings of the CVPR*, 2023, pp. 23 171–23 181.
[13] N. Hirose *et al.*, "Lelan: Learning a language-conditioned navigation policy from in-the-wild video," in *CoRL*. PMLR, 2025, pp. 666–688.
[14] C. Glossop *et al.*, "Cast: Counterfactual labels improve instruction following in vision-language-action models," *arXiv:2508.13446*, 2025.
[15] A.-C. Cheng *et al.*, "Navila: Legged robot vision-language-action model for navigation," *arXiv:2412.04453*, 2024.
[16] J. Gu *et al.*, "Vision-and-language navigation: A survey of tasks, methods, and future directions," *arXiv:2203.12667*, 2022.
[17] D. Shah *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *CoRL*, 2023, pp. 492–504.
[18] O. Mees *et al.*, "What matters in language conditioned robotic imitation learning over unstructured data," *RA-Letters*, vol. 7, no. 4, pp. 11 205–11 212, 2022.
[19] A. J. Sathyamoorthy *et al.*, "Convoi: Context-aware navigation using vision language models in outdoor and indoor environments," *arXiv:2403.15637*, 2024.
[20] Z. Xu *et al.*, "Mobility vla: Multimodal instruction navigation with long-context vlms and topological graphs," in *CoRL*, 2024.
[21] S. Reed *et al.*, "A generalist agent," *arXiv:2205.06175*, 2022.
[22] D. Driess *et al.*, "Palm-e: an embodied multimodal language model," in *Proceedings of the ICML*, 2023, pp. 8469–8488.
[23] O. M. Team *et al.*, "Octo: An open-source generalist robot policy," *arXiv:2405.12213*, 2024.
[24] K. Black *et al.*, "$\pi_0$: A vision-language-action flow model for general robot control," *arXiv:2410.24164*, 2024.
[25] C. Lynch *et al.*, "Learning latent plans from play," *arXiv:1903.01973*, 2019.
[26] R. Doshi *et al.*, "Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation," *arXiv:2408.11812*, 2024.
[27] V. Myers *et al.*, "Goal representations for instruction following: A semi-supervised language interface to control," in *CoRL*. PMLR, 2023, pp. 3894–3908.
[28] FrodoBots, "Frodobots-2k," 2024. [Online]. Available: https://huggingface.co/datasets/frodobots/FrodoBots-2K
[29] H. Xu *et al.*, "End-to-end learning of driving models from large-scale video datasets," in *Proceedings of the CVPR*, 2017.
[30] M. J. Kim *et al.*, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv:2502.19645*, 2025.
[31] N. Savinov *et al.*, "Semi-parametric topological memory for navigation," *arXiv:1803.00653*, 2018.
[32] N. Hirose *et al.*, "Deep visual mpc-policy learning for navigation," *RA-Letters*, vol. 4, no. 4, pp. 3184–3191, 2019.
[33] "EarthRover Zero," https://shop.frodobots.com/collections/earth-rovers/products/earthroverzero, Accessed: 2025-09-15.
[34] T. Niwa *et al.*, "Spatio-temporal graph localization networks for image-based navigation," in *Proceedings of the IROS*. IEEE, 2022, pp. 3279–3286.
[35] "Go1," https://shop.unitree.com/products/, Accessed: 2025-09-15.
[36] M. Minderer *et al.*, "Simple open-vocabulary object detection," in *Proceedings of the ECCV*. Springer, 2022, pp. 728–755.
[37] N. Hirose and K. Tahara, "Depth360: Self-supervised learning for monocular depth estimation using learnable camera distortion model," in *Proceedings of the IROS*. IEEE, 2022, pp. 317–324.
[38] S. Belkhale and D. Sadigh, "Minivla: A better vla with a smaller footprint," 2024. [Online]. Available: https://github.com/Stanford-ILIAD/openvla-mini
[39] M. Shukor *et al.*, "Smolvla: A vision-language-action model for affordable and efficient robotics," *arXiv:2506.01844*, 2025.
[40] D. Shah *et al.*, "Rapid exploration for open-world navigation with latent goal models," *arXiv:2104.05859*, 2021.
[41] G. Kahn *et al.*, "Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation," in *Proceedings of the ICRA*. IEEE, 2018, pp. 5129–5136.

[42] S. Triest *et al.*, "Tartandrive: A large-scale dataset for learning off-road dynamics models," in *ICRA*, 2022, pp. 2546–2552.
[43] A. Shaban *et al.*, "Semantic terrain classification for off-road autonomous driving," in *CoRL*. PMLR, 2022, pp. 619–629.
[44] H. Karnan *et al.*, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," *RA-Letters*, vol. 7, no. 4, pp. 11 807–11 814, 2022.
[45] N. Hirose *et al.*, "Sacson: Scalable autonomous control for social navigation," *RA-Letters*, vol. 9, no. 1, pp. 49–56, 2023.
[46] ——, "Gonet: A semi-supervised deep learning approach for traversability estimation," in *IROS*, 2018, pp. 3044–3051.

# APPENDIX

## A. OmniVLA-edge based on vision-based navigation policies

In addition to our VLA-based architecture, we design another network, OmniVLA-edge (Fig.6), based on prior single-modality vision-based navigation policies such as ViNT[8], NoMaD [9], MBRA [11], and LeLaN [13]. Building on ViNT for egocentric goal image-conditioned navigation, we add a projector for 2D goal-pose conditioning and ResNet and CLIP networks with FiLM for language prompt conditioning [13, 26]. Similar to the VLA-based model, we design an attention mask according to the selected modality $t_m$ during data sampling. Unlike the VLA-based network, this model employs early fusion, conditioning tokens on each modality before feeding them into the transformer. To maintain temporal consistency with a lightweight architecture, we feed tokens from the last $M = 5$ image steps. Following NoMaD [9], we compute the mean of the transformer-generated tokens and pass them to the action head to produce actions $\{\hat{a}_i\}_{i=1\ldots N}$. Pre-trained weights are used for the EfficientNet-B0, ResNet, and CLIP elements as shown in Fig. 6.

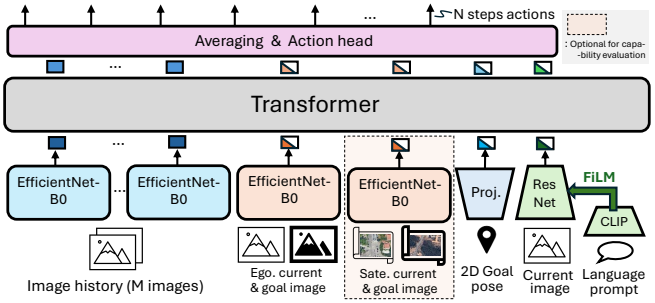



**Fig. 6: Network architecture of OmniVLA-edge based on the vision-based navigation policies.**

## B. Reannotation model for BDD-V dataset

BDD-V [29] consists of observations captured by a smartphone mounted on a car dashboard, paired with GPS signals used as actions in the original paper. Compared to other datasets, BDD-V is larger and covers more diverse environments. However, directly using GPS-based actions for training is challenging: they are imprecise due to GPS uncertainty, collected at 1 Hz (vs. 3 Hz for other datasets), and the vehicle speed is about 40 times faster than the mobile robots used in other datasets (0.5 m/s).

To address this embodiment gap, we adapt the reannotation approach in [11] and train a version of the MBRA model to generate feasible trajectories for our target robots. The original MBRA penalizes virtual collisions using estimated 3D points, but BDD-V images often include the dashboard, whose appearance varies with camera pose and can cause false collisions, trapping the model. We resolve this by training MBRA on a combination of GNM and BDD-V, applying the collision-avoidance objective only to GNM. This allows the model to implicitly learn collision avoidance from GNM while adapting to BDD-V's visual distribution. Additionally, we constrain linear and angular velocities to 0.0–0.5 m/s and ±1.0 rad/s, and enforce the kinematic model of a coaxial two-axle robot, ensuring generated trajectories are consistent with other datasets.

## C. Pre-trained checkpoints and backbones for the VLAs

Table VI lists the language and visual backbones for versions of OmniVLA and several VLA baselines. These models span a range of sizes and architectures, allowing us to evaluate how pre-trained models contribute prior knowledge to learning OmniVLA. Further details on the pre-trained checkpoints and backbones are in the original papers.

| Model name | Language backbone | Visual backbone | Model size |
|---|---|---|---|
| SmolVLA [39] | SmolLM2 | SigLIP | 500M |
| MiniVLA [38] | Qwen2.5-0.5B | DINOv2+SigLIP | 1.0B |
| CounterfactualVLA [14] | Gemma-2B | SigLIP | 2.9B |
| OmniVLA-edge | CLIP | EfficientNet-B0 | 50M |
| OmniVLA | Llama2-7B | DINOv2+SigLIP | 7.5B |

**TABLE VI:** We compare two variants of OmniVLA with VLA baselines: the Gemma-based CounterfactualVLA and the omni-modal conditioning recipe applied to SmolVLA and MiniVLA.

## D. Breakdown of GNM and LeLaN dataset mixture

The GNM mixture in Table I comprises seven publicly available datasets—RECON [40], CoryHall [41], Tartan-Drive [42], Seattle [43], SCAND [44], SACSoN [45], and GO Stanford 4 [32]—combined as in [6]. Using the curated expert actions as both action commands and 2D goal poses, OmniVLA learns policies conditioned on egocentric images, 2D poses, or their combinations.

LeLaN [13] provides synthetic action labels, language prompts, and object poses from SACSoN, GO Stanford 2 & 4 [32, 46], HumanWalk, and YouTube videos. We leverage these annotations to train OmniVLA for language grounding.

## E. Objective design

Following [13], we introduce the additional objectives $J_{obj}$ and $J_{sm}$ in addition to the main objective $J_{il}$ and train our OmniVLA policy, $\pi_\theta$ to minimize the entire $J$.

$$\min_\theta J := J_{il} + m_{obj} J_{obj} + J_{sm}, \tag{1}$$

$J_{obj}$ is designed as $(p_{obj} - \hat{a}_N)^2$ to encourage the policy to generate actions that move toward the target object pose $p_{obj}$ in language-conditioned navigation. Following [13], we penalize the $N$-th action $\hat{a}_N$ to be close to $p_{obj}$. Since $J_{obj}$ is only for the LeLaN dataset to learn language grounding, we set $m_{obj} = 1$ for the LeLaN dataset, otherwise $m_{obj} = 0$ to mask out $J_{obj}$. In addition, $J_{sm} = \frac{1}{N-1}\sum_{i=1}^{N-1}(\hat{a}_{i+1} - \hat{a}_i)^2$ is the objective to minimize the action deltas for regularization.