

# 412 Appendix

## 413 A Additional Experiment in Home Environment

414 We conduct comparative experiments in a home environment to evaluate our method in more diverse  
415 and challenging environments. We select the home, which is not included in all datasets. In the  
416 home, we choose ten objects, make five prompts for each object, and navigate the robot toward the  
417 target object, similar to our evaluation in the main paper. The appearance of the target objects and  
418 the given prompts are shown in the Appendix below.

419 The two baselines in the home perform better than the office environments. However, there is still  
420 an explicit advantage between our method and the baselines. Our method trained on augmented  
421 YouTube videos learns a general policy that can navigate towards novel objects in novel environ-  
422 ments. Here, “novel” indicates that the training dataset does not include images of the object in  
423 question in the environment seen during evaluation.

More evaluations in novel environments are shown in our supplemental material video.

Table 3: **Quantitative results using a prototype real robot in home environment.** We show the goal success rate. A success if determined by the robot reaching within a 0.2 [m] radius of the target object.

Method	Total	Simple prompts	Noisy prompts	Multiple objects
CoW	0.72	0.80	0.67	0.53
Owl-ViT + ViNT	0.60	0.60	0.60	0.40
Our method	<b>0.84</b>	<b>0.85</b>	<b>0.83</b>	<b>0.80</b>

424

## 425 B Additional Data Ablation

426 In addition to the robot dataset ablation study in Fig. 6, we conduct an additional dataset abla-  
427 tion study for the YouTube Tour Dataset and our Human-walking Dataset. By including more data  
428 sources in our augmented dataset, the performance of the trained language-conditioned navigation  
429 policy improves. We show an improvement in the performance of the policy by adding the In-  
430 door Navigation Dataset and our Human-Walking Dataset. We hypothesize that improvements from  
431 adding more data from YouTube saturate the least as due to the broad distribution of environments  
432 and objects within the dataset. We can use YouTube video data because of our data augmentation  
433 approach, which enables us to leverage the diverse in-the-wild video. Note that we add the test  
434 dataset to the YouTube Tour Dataset and Human-Walking Dataset due to make the balance of data  
between the three sources more even Fig. 7.

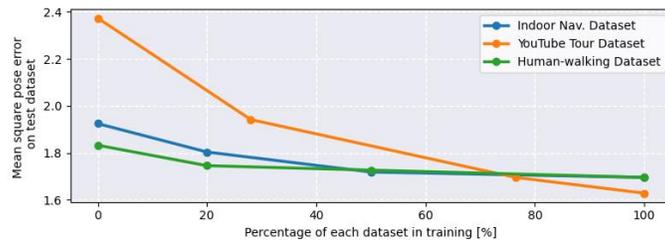


Figure 7: **Data Ablation.** An ablation of the percent of each dataset included in training data mixture, while keeping the entirety of the other datasets in the data mixture. The data ablation that studies the Indoor Navigation Dataset and Human-Walking Dataset use 76% of the YouTube dataset due to the addition of new YouTube data during the course of the project.

435

436 **C Model Ablations**

437 We also study how ablations of our model architecture impact the performance of our policy while  
 438 training on the LeLaN dataset. For the visual encoder, we replace "ResNet-FiLM" in our method  
 439 with "ViT-B32" and "ViT-ResNet50" of the pre-trained CLIP. For the text encoder, we employ larger  
 440 pre-trained CLIP text encoder "ViT-ResNet50" instead of "ViT-B32".

441 In this ablation study, we evaluate each model on the test dataset. Similar to the objective in training,  
 442 we calculate the mean square error between the generated virtual robot pose from the control pol-  
 443 icy and the target object pose. From Table 4, the pre-trained visual encoders from CLIP are worse  
 444 than the ResNet-FiLM trained on our augmented dataset. The visual features from the CLIP visual  
 445 encoder are insufficient to derive time-series velocity commands because they do not include geo-  
 446 metric information. Furthermore, the ResNet-FiLM inserts the text features from the text encoder  
 447 for low-level visual features, which helps to understand the target objects in the image view. In  
 448 addition, the larger CLIP text encoder helps with learning a precise control policy. However, the  
 449 advantage of a larger encoder is not significant on the test dataset. Furthermore, when navigating  
 450 with a real robot, its difference was trivial and, in fact, increased the computational load on the robot  
 451 controller. This not only reduced the frame rate, but also increased battery consumption. Therefore,  
 452 we used the the pre-trained text encoder from the "ViT-B32" CLIP model for the main model in the  
 paper.

Table 4: **Ablation study of our model architecture.** We use the pre-trained weights of ViT-B32 and ViT-ResNet50 from CLIP for both the visual and text encoders. When using ResNet-FiLM, we train our model from scratch.

Visual encoder			Text encoder		MSE
ResNet-FiLM	ViT-B32	ViT-ResNet50	ViT-B32	ViT-ResNet50	
✓			✓		1.291
✓				✓	1.202
	✓		✓		1.690
		✓		✓	1.673

453

454 **D Target Objects and Prompts in Evaluation**

455 In our evaluation, we select 18 objects in the university campus environment (inside and outside)  
 456 and 10 additional objects in the home environment and prompt the robot to navigate towards the  
 457 target objects. For each object, we feed 5 or 6 trials with different prompts (some of which are  
 458 noisy) and evaluate the robustness of the policy. Here we show the overview of the target objects  
 459 and the prompts in our evaluation. First 18 objects are from the university campus. The rest are from  
 460 a home environment.

461 First, two prompts for each object are for the simple prompts and the others are for noisy prompts,  
 462 which includes wrong adjectives (red) long prompts, or the prompts without the target object's  
 463 noun. The red border in the image indicates the presence of multiple corresponding objects in the  
 464 experimental environment. If the objects can be distinguished by prompts, success is considered only  
 465 if the robot reaches the correct object; if the objects cannot be distinguished by prompts, success is  
 considered if the robot reaches one of the objects that fits the description of the prompt.



- couch
- small rounded purple couch
- purple carpet
- round plush piece of furniture
- purple couch on the grey concrete floor, next to the blue grey pillow
- pinkish purple cushion-like stool



- fridges
- shiny fridge with a water dispenser
- arched fridge with a water server
- tall shiny appliance with two doors
- Fridge next to the wooden shelf
- dark silver something to storage food

466



- office chair
- grey and white swivel chair with arm rests and wheels
- **metal** chair with black arm rests
- **white** and grey structure with a sleek, ergonomic design
- **white** and grey something to sit on it
- **white** and grey chair with black suitcase



- closed door
- grey brown door with a black handle
- black door with a **silver** bar
- grey wooden surface with a black part
- **wooden** door in the white paneled wall
- grey panel



- painting
- colorful small painting
- painting of vegetables
- **vibrant canvas**
- painting on the plain **white** wall
- small poster



- sheep doll
- white stuffed sheep doll
- stuffed **cow**
- fluffy toy
- sheep doll sitting on the couch
- realistic sheep



- person
- person wearing black pants and black T shirts
- person wearing dark color pants and **grey** T shirts
- [first name], [family name]
- person sitting in the office chair



- whiteboard
- whiteboard with text written on it
- **clean** whiteboard
- reflective surface with text
- whiteboard with green box
- reflective surface with **blue** trash box



- stairs
- concrete stairs with black bands
- **spiral** stairs
- steps that lead to the next floor
- stairs with the metal railings



- doors
- white double doors
- **arched** doors
- white entry way
- doors next to the garbage



- stool
- thin metal stool
- **short** metal stool
- thin metal and wood piece of furniture
- stool next to the white table
- something to sit on it



- pillar
- concrete pillar
- **round** pillar
- supporting rectangular structure
- pillar with flyers and papers on it



- hand sanitizer
- white hand sanitizer station
- **curved** hand sanitizer post
- thin white post
- hand sanitizer next to the white wall.



- garbage bin
- blue garbage bin
- **round** trash bin
- grey box with labels on it
- garbage bin with green plants



- lamp post
- tall thin lamp post
- pole
- thin sleek structure
- lamp post next to the trees and rocks



- tree
- short green tree
- tall tree with **autumn** colors
- **large** plant with lots of leaves
- tree next to the dark rocks



- bench
- wooden long bench
- **metal curved** bench
- wooden seating area
- bench next to the green plants



- yellow sign
- yellow triangular sign
- yellow sign on the wall
- folded yellow structure with text on it
- yellow sign next to the concrete pillar



- bicycle
- black bicycle
- **electric** bikes
- vehicle with two wheels
- bicycle with the green plants



- dumpster
- green dumpster with black lid
- **curved** dumpster
- box with text 3yd-3011 on it
- dumpster along the concrete wall



- black car
- black minivan
- modern sleek black minivan
- Toyota Sienna
- black car with four wheels



- shoes shelf
- wooden shoes shelf
- **round** shoes shelf
- wooden furniture
- wooden shelf with many shoes



- toilet
- white toilet
- **square** toilet
- white something for peeing
- toilet next to the bathtub



- orange toy box
- sleek orange box with toys
- **wooden** orange bin filled with toys
- bin filled with something for kids
- orange box, next to the kitchen toy



- double door
- closed double door
- **curved** double door
- entry way to the closet
- double door with two black knobs



- backpack
- black backpack
- dark color backpack
- sleek **square** backpack
- black backpack with blue shiny band



- TV
- black display to see the video
- **round** TV
- electric device to see the movie
- large black TV along white wall



- plant pod
- wooden plant pod
- brown **plastic** plant pod
- box for growing plants
- wooden box with green plants

Figure 8: Overview of 28 target objects and various prompts in our evaluation.

## 467 **E Baseline Method**

468 In our evaluation, we conduct a comparative evaluation with two strong baselines trained on the  
469 internet scale datasets. Here we explain the details of each of their implementations.

470 **CLIP on Wheels (CoW)** We implement the best-performing CoW baseline with the OWL-ViT  
471 B/32 detector [51]. Similar to our method, we feed the current observation and the prompts corre-  
472 sponding to the target object into the OWL-ViT B/32 detector [51], which was trained an internet  
473 scale dataset to estimate the boundary box for the target object. We crop the estimated point clouds  
474 by the estimated boundary box and take the median value as the target object pose. To have a fair  
475 comparison without method using a single camera image only (no depth camera and LiDAR), we  
476 estimate the depth with Depth360 [45] and project it to estimate the point clouds. To control the  
477 robot toward the detected object, we use a state lattice motion planner to generate the linear and  
478 angular velocity commands. The details of the implemted state lattice motion planning are shown  
479 in this appendix below. We limit the scope of instructions to object navigation for objects within  
480 view from the starting point of the robot trajectory. Therefore, we do not implement the exploration  
481 portion of CoW.

482 **OWL-ViT + ViNT** To compare our method with a learning-based method, we leverage the founda-  
483 tion model for the vision-based navigation, which can navigate the robot towards a goal position  
484 conditioned on a goal image view. To take a goal image view corresponding a target object, this  
485 baseline leverages Owl-ViT, a VLM trained on internet-scale data and combine it with ViNT, a con-  
486 trol policy trained on multiple dataset collected by various mobile robots. Specifically, we feed the  
487 cropped image from the Owl-ViT into the ViNT as a goal image.

## 488 **F Implementation details**

489 We show the details of our training and the evaluation setup using a real prototype robot in language-  
490 conditioned navigation.

### 491 **F.1 Training**

492 To train our control policy, we randomly choose 256 observations from our whole dataset. Since  
493 one observation contains multiple objects and each object contains multiple prompts, in almost all  
494 cases, we randomly select the object and prompt (which is based on the object).

495 By feeding the observations and the prompts into the model, we calculate our model and generate  
496 a sequence of the velocity commands for  $N(=24)$  steps. Then, we estimate the virtual robot pose  
497  $N$  steps in the future via our kinematic model (integration of the velocity commands in our case).  
498 Finally, we calculate the objective  $J$  in Eqn. 1 and update our policy  $\pi_\theta$ . Our training is with an  
499 Adam optimizer using a learning rate 0.0001 on a workstation with a Intel i9 CPU, 96GB RAM and  
500 an NVIDIA RTX 4090 GPU.

### 501 **F.2 Robot experiment**

502 Figure 9 shows the overview of a prototype mobile robot in navigation. We calculate the control  
503 policy on the edge robot controller, Nvidia Orin AGX, with the best frame rate for each method. We  
504 mount the omnidirectional camera, a RICOH Theta S on the robot and only use the front-side fisheye  
505 camera as the observation. Since we learn the visual encoder from scratch, there are no restrictions  
506 on the camera on the robot, but we use cameras with a wide FOV to reduce blind spots and make  
507 object detection easier.

508 In the evaluation, we provide the language instruction to the policy once at the beginning of naviga-  
509 tion to reduce the computational load in each step. To control the real robot, we repeatedly calculate  
510 our control policy at the best frame rate on the robot edge controller and feed the first step veloc-

511 ity command  $\{v_0, \omega_0\}$  in the generated sequence of the velocity commands  $\{v_i, \omega_i\}_{i=0 \dots N}$ , similar  
 512 to the receding horizon control. We test our method against two baselines, CLIP on Wheels and  
 OWL-ViT + ViNT.

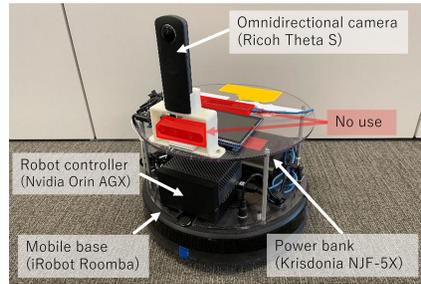


Figure 9: **Overview of the prototype mobile robot.** Note that we only use the front side camera of the omnidirectional camera, Ricoh Theta S, to navigate the robot.

513

## 514 G YouTube Video List

515 We list all URLs of the YouTube video in our YouTube Tour Dataset below.

- 516 • [https://www.youtube.com/watch?v=vQU\\_QydOUIw](https://www.youtube.com/watch?v=vQU_QydOUIw)
- 517 • <https://www.youtube.com/watch?v=5J2Wsvnk-Ec>
- 518 • <https://www.youtube.com/watch?v=b9thcSOI8bw>
- 519 • <https://www.youtube.com/watch?v=V511PNMx2uw>
- 520 • [https://www.youtube.com/watch?v=DO-JDTu\\_h5I](https://www.youtube.com/watch?v=DO-JDTu_h5I)
- 521 • <https://www.youtube.com/watch?v=oQ61ijCHego>
- 522 • <https://www.youtube.com/watch?v=EwJQG74b174>
- 523 • <https://www.youtube.com/watch?v=rM01DH0bv1o>
- 524 • <https://www.youtube.com/watch?v=9MWWZeCr3QE>
- 525 • <https://www.youtube.com/watch?v=-3vt2Mylvsw>
- 526 • <https://www.youtube.com/watch?v=HknDp84cFBM>
- 527 • [https://www.youtube.com/watch?v=l\\_s9YAluXBY](https://www.youtube.com/watch?v=l_s9YAluXBY)
- 528 • <https://www.youtube.com/watch?v=k3Q1vse7In8>
- 529 • <https://www.youtube.com/watch?v=ISNDJ2Pjq34>
- 530 • [https://www.youtube.com/watch?v=4jDa\\_5S-0W4](https://www.youtube.com/watch?v=4jDa_5S-0W4)
- 531 • [https://www.youtube.com/watch?v=2O7JrGu\\_mVk](https://www.youtube.com/watch?v=2O7JrGu_mVk)
- 532 • <https://www.youtube.com/watch?v=je8267s9z38>
- 533 • <https://www.youtube.com/watch?v=tWovplr-ois>
- 534 • <https://www.youtube.com/watch?v=UmCbkpRUOA4>
- 535 • <https://www.youtube.com/watch?v=Ea2yExKlg7w>
- 536 • <https://www.youtube.com/watch?v=1Zu6Xct5bLQ>
- 537 • <https://www.youtube.com/watch?v=9IluzedLtYs>
- 538 • [https://www.youtube.com/watch?v=lnYfw\\_ryOdQ](https://www.youtube.com/watch?v=lnYfw_ryOdQ)
- 539 • <https://www.youtube.com/watch?v=9r5eK5JXzLo>
- 540 • <https://www.youtube.com/watch?v=LdWHy-f3jYg>
- 541 • <https://www.youtube.com/watch?v=Kcc7zuQDlpE>
- 542 • <https://www.youtube.com/watch?v=r-98ADAXxQM>
- 543 • <https://www.youtube.com/watch?v=iRfQa2SEu0Q>
- 544 • <https://www.youtube.com/watch?v=NzFbFARYhfE>
- 545 • <https://www.youtube.com/watch?v=i3QkZ0xW92Y>
- 546 • <https://www.youtube.com/watch?v=stUYODYcPCI>
- 547 • <https://www.youtube.com/watch?v=GCKYfI5LRYM>
- 548 • <https://www.youtube.com/watch?v=848EpwPmQfA>
- 549 • <https://www.youtube.com/watch?v=Bq4rmeIvJbs>

- 550 • <https://www.youtube.com/watch?v=uVy9TKMA-f8>
- 551 • [https://www.youtube.com/watch?v=\\_OZhGsKdBY](https://www.youtube.com/watch?v=_OZhGsKdBY)

## 552 H State lattice motion planning

553 We implemented sampling-based motion planning as the local motion planner in **CLIP on Wheels**  
 554 **(CoW)** baseline. We generated 15 motion primitives assuming steady linear and angular velocity  
 555 commands for 8 steps (2.664 s). The pairs of linear and angular velocity commands are  $(v_s, \omega_s) =$   
 556  $(0.0, 0.0), (0.2, 0.0), (0.2, 0.3), (0.2, 0.6), (0.2, 0.9), (0.2, -0.3), (0.2, -0.6), (0.2, -0.9), (0.5, 0.0),$   
 557  $(0.5, 0.3), (0.5, 0.6), (0.5, 0.9), (0.5, -0.3), (0.5, -0.6), (0.5, -0.9)$ . We selected these 15 motion  
 558 primitives by balancing computational load and navigation performance.

559 By integrating these velocity commands for 8 steps, we obtained 15 trajectories such as  
 560  $\{\{^s p_i^j\}_{i=1\dots 8}\}_{j=1\dots 15}$ , where  $^s p_i^j$  is the  $i$ -th virtual robot pose on the  $j$ -th motion primitive. To  
 561 select the best motion primitive, we calculated the following cost value for each primitive.

$$J_s^j = \min_i (\hat{p}_{\text{Obj}} - ^s p_i^j)^2 \quad (2)$$

562 Here,  $\hat{p}_{\text{Obj}}$  indicates the estimated target object pose in **CLIP on Wheels (CoW)** baseline. This  
 563 objective calculates the squared errors between all 8 poses in the  $j$ -th motion primitive and the goal  
 564 pose and selects the minimum one to evaluate the goal-reaching performance. Then, we choose the  
 565 motion primitive with the minimum  $\{J_s^j\}_{j=1\dots 15}$  and assign the corresponding velocity commands  
 566  $v_s$  and  $\omega_s$  to control the robot during navigation.