ComposableNav: Instruction-Following Navigation in Dynamic Environments via Composable Diffusion

Zichao Hu[1], Chen Tang[1], Michael J. Munje[1], Yifeng Zhu[1], Alex Liu[1], Shuijing Liu[1], Garrett Warnell[1,2], Peter Stone[1,3], Joydeep Biswas[1,4]

***Abstract*— We study how robots navigate dynamic environments while following instructions. Unlike prior work where the instructions only specify the navigation goals in static environments, our work focuses on instructions that specify robot behaviors (*e.g.,* "yield to a pedestrian"). This problem poses two key challenges: (1) the robot must learn to satisfy an exponential number of specification combinations across different instructions, and (2) the robot must reasoning about multiple specifications concurrently rather than processing them sequentially when operating in dynamic environments. To address these challenges, we propose ComposableNav, based on the insight that following an instruction amounts to independently satisfying its constituent specifications, each satisfied by a different motion primitive. ComposableNav uses diffusion models to individually learn these primitives and composes them in parallel at deployment to generate an instruction-following trajectory. For example, "overtake the pedestrian in front and stay on the sidewalk" is achieved by composing the primitives "overtake the pedestrian" and "stay on the sidewalk." In addition, we introduce a two-stage training procedure consisting of supervised pre-training followed by reinforcement learning fine-tuning, enabling effective learning of each motion primitive without requiring primitive-specific demonstrations. Through both simulation and real-world experiments, we show that ComposableNav enables robots to follow a broad range of instructions and significantly outperforms both non-compositional VLM-based policies and baselines that compose costmaps.**

I. Introduction

Developing robots that can effectively navigate by following instructions has been a long-standing goal in robotics research. Existing work has predominantly tackled the instruction-following navigation problem in *static* environments [1], [2], [3], with instructions specifying the navigation goals. In contrast, we focus on instruction-following navigation in *dynamic* environments. Specifically, we consider the under-explored settings where the instructions describe specific robot behaviors (*e.g.,* "yield to a pedestrian") with respect to the other dynamic obstacles or agents in the environments. Addressing this problem requires developing methods capable of grounding high-level instructions into fine-grained, low-level actions that account for the dynamic behaviors of other agents. Solving this problem would allow end users (human or AI agents) to customize robotic behaviors beyond their default settings, in ways that align with user preferences and nuanced social interactions.

[1]Department of Computer Science, The University of Texas at Austin, Austin, TX, USA. Correspondence: zichao@utexas.edu

[2]Garrett Warnell is also with DEVCOM Army Research Laboratory.

[3]Joydeep Biswas is also with NVIDIA.

[4]Peter Stone is also with Sony AI.

Two major challenges arise when addressing instruction-following navigation in dynamic environments. First, a single instruction may contain multiple specifications for the robot to follow, and as the robot's capabilities expand, the potential combinations of these specifications grow exponentially. This exponential increase makes popular learning-based methods, such as imitation learning [4] or reinforcement learning [5], [6], impractical as they demand substantial data and computational resources. Second, in dynamic environments, the effects of individual specifications become interleaved along the robot's trajectory, and multiple specifications can simultaneously influence the robot's behavior. This challenge requires methods to reason about multiple specifications in parallel, rather than treating each one in sequence [7], [8]. With the advancements of large vision language models (VLMs), leveraging the strong reasoning capabilities of these VLMs may appear to be promising. However, these methods struggle with fine-grained control to align the robot behaviors with instructions in dynamic environments [9], [10], [11].

To address these challenges, we build our solution upon the idea of *composition*: following an instruction often amounts to independently satisfying each of its constituent specifications. For example, the instruction "overtake the pedestrian in front and stay on the sidewalk" can be decomposed into two specifications: "overtake the pedestrian" and "stay on the sidewalk." This insight allows us to simplify the problem. Instead of training a single model to handle all possible combinations of specifications—which can grow exponentially—we train individual motion primitives for each specification. At deployment time, we compose the relevant primitives in parallel based on the instruction. This approach reduces complexity, requiring only a number of primitives that scales linearly with the number of specification types.

We propose **ComposableNav**, a composable, diffusion-based motion planner. The key insight behind our approach is that diffusion models [12], [13] can effectively represent complex probability distributions and multiple models can be composed to form a joint distribution. This enables us to model each motion primitive as a distribution over trajectories that satisfy a instruction specification. At deployment time, a trajectory is sampled from the composed joint distribution, resulting in behavior that simultaneously satisfies all instruction specifications. To learn each motion primitive, we introduce a two-stage training procedure consisting of supervised pre-training followed by reinforcement learning fine-tuning. This approach addresses the challenge of the lack of demonstration data for individual primitives.

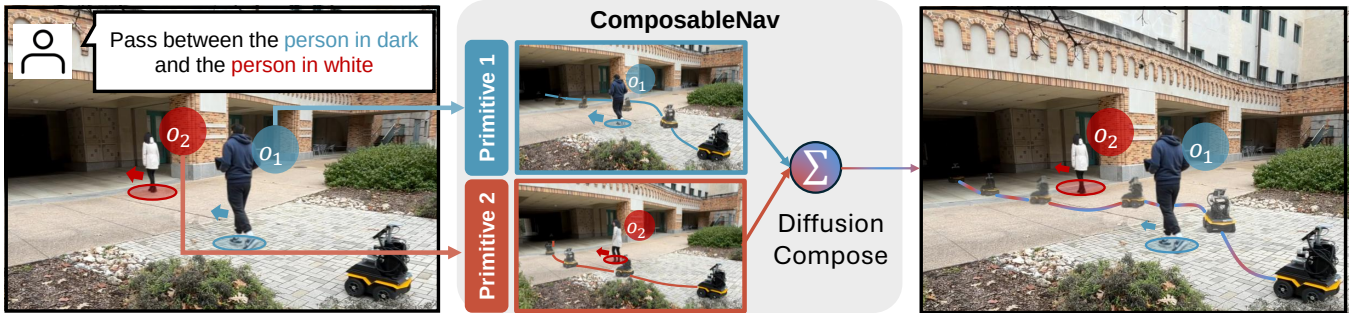


Fig. 1: **Compose motion primitives for the Instruction-Following Navigation in Dynamic Environments problem.** Given an instruction that specifies how a robot should interact with entities in the scene, ComposableNav leverages the composability of diffusion models to compose motion primitives to generate instruction-following trajectories.

Finally, to ensure real-time performance, we incorporate a model predictive controller (MPC) [14] for low-latency action execution and introduce a fast replanning strategy.

We demonstrate the effectiveness of ComposableNav through experiments in both simulation and the real world. With just six motion primitives (See Table I for the instruction list), we build a testbed with 24 different scenarios composed from the atomic instructions representing different instructions. We show that ComposableNav excels at following unseen instructions compared to baseline approaches.

Our main contributions are summarized as follows:

1) We introduce the use of *composition* as a strategy for instruction-following navigation in dynamic environments, making the problem more tractable under limited training data and computational resources.
2) We propose a diffusion-based learning method to model motion primitives as probability distributions, enabling their composition at deployment time.
3) We develop a two-stage training procedure, combining supervised pre-training and reinforcement learning fine-tuning, that effectively learns motion primitives without the need for specialized demonstration datasets for each primitive.

II. Problem Formulation

We consider the problem of instruction-following robot navigation in dynamic environments, where the objective is to generate a motion trajectory τ that follows a given instruction I, based on the robot's observation O of the environment. We represent the motion trajectory τ as a sequence of 2D waypoints at fixed-time intervals, which are then tracked by a model predictive controller to produce fine-grained actions in real time. The observation O encodes the state of entities relevant to the instruction, such as the current and predicted positions of dynamic agents. Note that other representations are also possible, such as full SE(3) poses for τ or RGB images for O.

In this work, we assume an instruction I can be decomposed into a set of independent specifications $I \rightarrow \langle \phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(k)} \rangle$. Each specification $\phi^{(i)} : \tau \times O \rightarrow [0, 1]$ evaluates whether the trajectory meets the corresponding requirement, returning 1 if it does and 0 otherwise. To determine whether a trajectory τ follows an instruction I, τ must satisfy all relevant specifications. Formally,

$$\tau \text{ follows } I \text{ iff } \forall i \in [1, \cdots, k], \phi^{(i)}(O, \tau) = 1. \quad (1)$$

Solving this problem is challenging because the trajectory must simultaneously satisfy all specifications, whose combinations can grow exponentially. In the following sections, we explain how leveraging the diffusion models enables us to compose motion primitives and generate trajectories that can follow instructions during robot deployment.

III. Preliminaries

A. Conditional Diffusion Models

In this work, we consider conditional diffusion probabilistic models [12], [13], [15], which belong to a family of generative models trained to represent a conditional distribution $p(x \mid c)$, where c is the corresponding context. These models are trained to reverse a forward diffusion process $q(x_t \mid x_{t-1})$ that gradually adds Gaussian noise to the data $x_0 \sim p(x|c)$. To learn this reverse process, the model is trained to predict the noise at each step t using a denoising network, $f_\theta(x_t, t, c)$, where x_t is the noisy data at step t. The network is optimized using the following training objective:

$$\mathcal{L}_{\text{MSE}}(\theta) = \mathbb{E}_{x_0, \epsilon, t, c}\left[\|\epsilon - f_\theta(x_t, t, c)\|^2\right], \quad (2)$$

which is justified as maximizing a variational lower bound on the log-likelihood of the data [12].

At inference time, the model generates a data sample by starting from Gaussian noise $x_T \sim \mathcal{N}(0, \mathbf{I})$ and progressively denoising it using the learned denoising network for T steps. The reverse process at each timestep t follows a Gaussian distribution:

$$p_\theta(x_{t-1} \mid x_t, c) = \mathcal{N}(x_t - f_\theta(x_t, t, c), \sigma_t^2 \mathbf{I}), \quad (3)$$

where $\sigma_t^2 \mathbf{I}$ is a time-dependent covariance matrix treated as a hyperparameter [16]. This iterative process continues until a final sample x_0 is obtained, which approximates the true conditional distribution $p(x \mid c)$.

B. Denoising Diffusion Policy Optimization (DDPO)

ComposableNav follows the denoising diffusion policy optimization technique (DDPO) proposed by Black et al [17] to use RL to fine-tune diffusion models to generate the motion primitives corresponding to the atomic instructions. DDPO models the multi-step denoising process as a multi-step Markovian Decision Process (MDP), defined as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \rho_0, \mathcal{P}, R \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, ρ_0 is the distribution of initial states, $\mathcal{P}$ is the transition kernel, and R is the reward function. We denote the timestep of this multi-step MDP as i. The denoising process is mapped into this MDP as follows:

$$\begin{aligned} s_i &\triangleq \langle x_t, t, c \rangle \in \mathcal{S}, \\ a_i &\triangleq x_{t-1} \in \mathcal{A}, \\ \pi(a_i \mid s_i) &\triangleq p_\theta(x_{t-1} \mid x_t, c), \\ \rho_0(s_0) &\triangleq \langle \mathcal{N}(0, \mathbf{I}), \delta_T, p(c) \rangle, \\ P(s_{i+1} \mid s_i, a_i) &\triangleq \langle \delta_{x_{t-1}}, \delta_{t-1}, \delta_c \rangle, \\ R(s_i, a_i) &\triangleq \begin{cases} r(x_0, c) & \text{if } t = 0, \\ 0 & \text{otherwise}, \end{cases} \end{aligned} \tag{4}$$

where δ_y denotes the Dirac distribution with nonzero density only at y.

The key insight behind this technique is that the reverse process in a diffusion model is a Markovian process, where each denoising step $p_\theta(x_{t-1} \mid x_t, c)$ is modeled as a Gaussian distribution (see Eq. 3). By interpreting each denoising step as the policy $\pi(a_i \mid s_i)$ in an MDP, the policy itself becomes Gaussian, which allows for the exact evaluation of log-likelihoods and their gradients with respect to the diffusion model parameters. As a result, this formulation enables the use of policy gradient methods such as Proximal Policy Optimization (PPO) [18] to optimize the diffusion model's denoising network.

The DDPO algorithm alternates between (1) collecting denoising trajectories $\langle x_T, x_{T-1}, \ldots, x_0 \rangle$ via sampling and (2) updating the model parameters using gradient descent. Finally, the policy gradient objective used in DDPO can be expressed as:

$$\mathcal{L} = \mathbb{E}\left[\sum_{t=1}^{T} \frac{p_\theta(x_{t-1} \mid x_t, c)}{p_{\theta_{\text{old}}}(x_{t-1} \mid x_t, c)} \nabla_\theta \log p_\theta(x_{t-1} \mid x_t, c) r(x_0, c) \right], \tag{5}$$

where the expectation is taken over denoising trajectories generated by the previous model parameters θ_{old}.

IV. ComposableNav

In this section, we introduce ComposableNav, a composable, diffusion-based motion planner. As shown in Fig. 2, ComposableNav learns and composes motion primitives from instructions to generate trajectories that satisfy the corresponding specifications. We begin by describing how ComposableNav learns individual motion primitives without requiring primitive-specific demonstration data in Sec. IV-A. Next, we explain how these primitives are composed to follow instructions in Sec. IV-B. Finally, we present additional techniques that enable ComposableNav to operate in real-time during deployment in Sec. IV-C.

A. Learning Motion Primitives Without Primitive-Specific Demonstration Data

Diffusion planners are typically trained in a supervised manner using large-scale demonstration datasets [19], [20], [21]. However, our problem lacks a specialized dataset for different robot motion primitives. It prevents us from following the common practice to train these motion primitives through supervised learning.

To address this problem, we leverage two key design choices. First, although we lack specialized datasets for individual motion primitives, it is relatively easy to obtain a general-purpose navigation dataset consisting of diverse, collision-free, and goal-reaching trajectories in dynamic environments — either from existing real-world datasets [22], [23] or simulation [24]. Such datasets allow us to pre-train a base diffusion planner to generate diverse and feasible trajectories across various environments. Second, inspired by denoising diffusion policy optimization (DDPO) [17], we adopt reinforcement learning (RL) techniques to fine-tune the pre-trained base model for different primitives given primitive-specific reward functions. The core intuition is that evaluating whether a trajectory aligns with an instruction specification (*e.g.,* using rule-based heuristics or black-box vision-language models (VLMs)) is often easier than directly collecting such a trajectory. Building on these insights, we propose a two-stage procedure involving supervised pretraining followed by RL fine-tuning.

Supervised Pre-training. To pre-train a base diffusion model, we first generate diverse trajectory data in simulation for simplicity and scalability. Following prior works [19], [24], we randomly synthesize environments with varying entities (*e.g.,* dynamic agents or terrain regions) and goal locations. We then use a geometric planner to generate a diverse set of collision-free, goal-reaching, and smooth trajectories. To account for dynamic environments, these trajectories must be *time-dependent*, so we employ a spatio-temporal Hybrid A* planner. In addition, we also want to capture the distribution of diverse feasible trajectories within the same environment (*e.g.,* both detouring left or right around an obstacle in front are feasible trajectories). Hence, we use a Rapidly-Exploring Random Tree planner to randomly generate candidate trajectories and then select waypoints along the trajectories as subgoals for Hybrid A* to track. We also vary the hyperparameters for the planners (*e.g.,* velocity cost) to further enhance trajectory diversity.

Once a diverse set of time-dependent trajectories is generated, we pretrain a base diffusion model via supervised learning, using the objective in Eq. 2. The model learns a conditional denoising network $f_\theta^{(\text{base})}(\tau_t, t, O)$, which predicts the noise ϵ to denoise the trajectory τ_t at step t, conditioned on environment observations O. We adopt an object-centric representation for the observations, encoding each entity separately and then using a transformer encoder

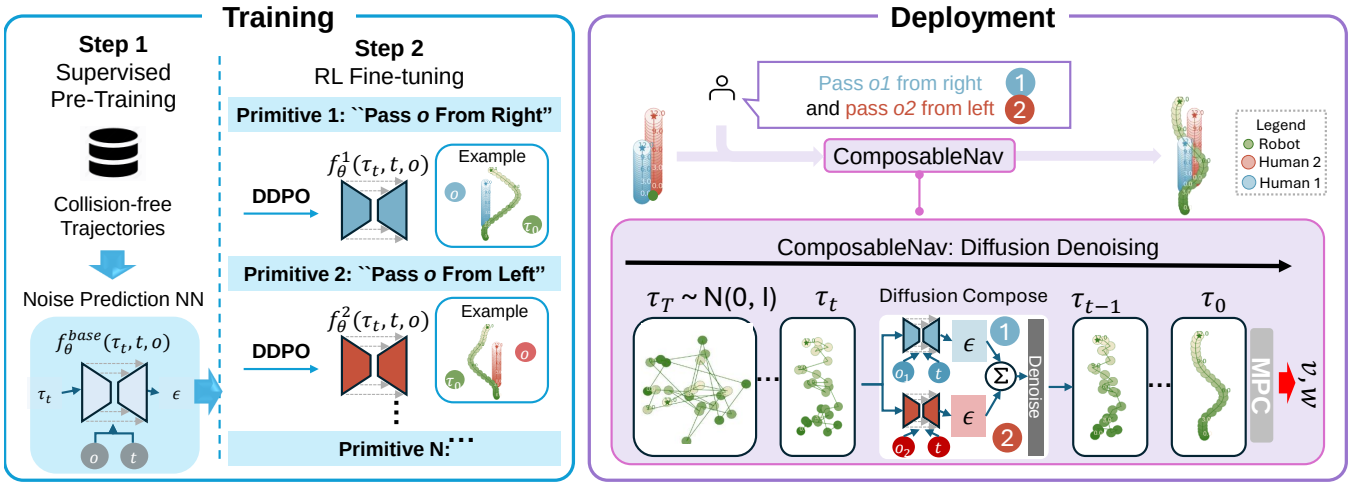


Fig. 2: **ComposableNav framework.** The training phase learns multiple motion primitives corresponding to different instruction specifications. It consists of a supervised pre-training stage followed by reinforcement learning fine-tuning. During deployment, ComposableNav composes the primitives based on the given instruction specifications using a diffusion-based denoising process to generate instruction-following navigation trajectories.

to attend over these embeddings to produce a global context feature. To handle varying trajectory lengths (*e.g.,* due to differing goal locations), we pad shorter trajectories with the final goal position to ensure uniform length during training.

RL Fine-tuning. We then fine-tune the base model *separately* for each motion primitive using RL, following the DDPO approach described in Sec. III-B. For each primitive, we randomly generate simulation environments containing only the entities relevant to the corresponding instruction specification. The diffusion model then generates trajectories for these environments, which are evaluated using a reward function based on how well they align with the instruction. While the reward function can take various forms (*e.g.,* black-box VLMs or RLHF-style reward models), we adopt a simple rule-based heuristic approach, as the primitives considered in this work are straightforward to evaluate. The resulting trajectories and rewards are stored in a replay buffer, and the model is updated using the Proximal Policy Optimization algorithm [18]. Finally, after fine-tuning, we obtain multiple diffusion models $f_\theta^{\phi^{(i)}}(\tau_t, t, O)$, each representing a motion primitive associated with a specification $\phi^{(i)}$. These models can be composed at deployment time to generate trajectories that follow more complex, unseen combinations of instructions.

B. Trajectory Generation via Composing Motion Primitives

ComposableNav generates an instruction-following motion trajectory τ by sampling from the conditional distribution $p(\tau|\phi^{(1)}, o^{(1)}, \cdots, \phi^{(k)}, o^{(k)})$, where each $\phi^{(i)}$ is a specification extracted from the instruction I, *i.e.,* $I \rightarrow \langle \phi^{(1)}, \cdots, \phi^{(k)} \rangle$, and each $o^{(i)}$ is the environment observation corresponding to $\phi^{(i)}$. We assume that both the specifications and the environment observations can be extracted using off-the-shelf large language models and vision foundation models; the details of these extraction processes are left as implementation details not discussed in this work.

Following prior work [25], we can factorize the conditional distribution as follows:

$$\begin{aligned} p(\tau|\phi^{(1)}, o^{(1)}, \cdots, \phi^{(k)}, o^{(k)}) & \\ \propto p(\tau, \phi^{(1)}, o^{(1)}, \cdots, \phi^{(k)}, o^{(k)}) & \\ = p(\tau) \prod_{i=1}^{k} p(\phi^{(i)}, o^{(i)} \mid \tau) & \\ \propto p(\tau) \prod_{i=1}^{k} \frac{p(\tau \mid \phi^{(i)}, o^{(i)})}{p(\tau)} & \end{aligned} \tag{6}$$

Here, each $p(\tau \mid \phi^{(i)}, o^{(i)})$ is represented by a motion primitive learned using a diffusion model with denoising network $f_\theta^{\phi^{(i)}}(\tau_t, t, o^{(i)})$. In contrast, $p(\tau)$ is the unconditioned motion primitive obtained by replacing the observation with a null input $\varnothing$, *i.e.,* $f_\theta^{\phi^{(i)}}(\tau_t, t, \varnothing)$, following the classifier-free guidance approach [15].

Based on Eq. 6, we compose motion primitives by summing the predicted noise outputs from the denoising networks for each specification, with the user-defined hyperparameter w_i controlling the guidance strength for the ith primitive [25], [19]. Specifically, we compute the composed noise $\hat{\epsilon}$ as:

$$\begin{aligned} \hat{\epsilon} = & \frac{1}{k} \sum_{i=1}^{i=k} f_\theta^{\phi^{(i)}}(\tau_t, t, \varnothing) \\ & + \sum_{i=1}^{k} w_i (f_\theta^{\phi^{(i)}}(\tau_t, t, o^{(i)}) - f_\theta^{\phi^{(i)}}(\tau_t, t, \varnothing)) \end{aligned} \tag{7}$$

Finally, ComposableNav generates trajectories by iteratively applying the reverse diffusion process. Starting from

TABLE I: Instruction Specifications for Navigation Motion Primitives

Motion Primitive (MP)	**Instruction Specification**
Pass a person from the left (L)	The robot should pass the person from the left side.
Pass a person from the right (R)	The robot should pass the person from the right side.
Follow behind a person (F)	The robot should stay in a specific region behind the person relative to the person's position.
Yield to a person (Y)	The robot should not cross the region in front of the person.
Walk through a region (W)	The robot's trajectory should overlap with the specified region.
Avoid walking through a region (A)	The robot's trajectory should not overlap with the specified region.

a noisy trajectory $\tau_T \sim \mathcal{N}(0, \mathbf{I})$, we follow Eq. 3:

$$p_{\text{compose}}(\tau_{t-1} \mid \tau_t, \phi^{(1)}, o^{(1)}, \cdots, \phi^{(k)}, o^{(k)}) = \mathcal{N}(\tau_t - \hat{\epsilon}, \sigma_t^2 \mathbf{I}), \tag{8}$$

and after T denoising steps, the process yields a trajectory τ_0 that is more likely to satisfy all specifications of the given instruction.

C. Real-time Deployment

To enable ComposableNav to run in real-time on a robot, we employ a model predictive controller (MPC) [14] to track the time-dependent trajectories generated by the composed diffusion models. During navigation, the MPC uses a kinematic model to predict the robot's future positions and minimizes the difference between these predictions and the target time-dependent trajectory over a short planning horizon. It also enforces constraints on acceleration and velocity to ensure that each control input is feasible and safe for execution.

To support real-time replanning, ComposableNav draws inspiration from adaptive online replanning methods [26]. The key insight is that the current planned trajectory is likely already close to a good solution. Instead of discarding the current trajectory and replanning from scratch, ComposableNav perturbs it by applying a few steps of the diffusion forward process, then partially denoises it to generate an updated trajectory conditioned on the latest observations. This procedure uses significantly fewer diffusion steps compared to full trajectory generation from scratch, allowing the system to efficiently replan in real-time.

V. Experiment

In this section, we present experiments and results to evaluate ComposableNav in simulation and real-world. We are guided by the following research questions:

1) Can ComposableNav learn individual motion primitives that satisfy each instruction specification without relying on demonstration data?
2) To what extent can ComposableNav compose motion primitives to generate trajectories that satisfy unseen combinations of specifications, in comparison to baseline approaches?
3) Can ComposableNav operate in real-time when deployed on a real-world robot and enable the robot to follow instructions in dynamic environments involving pedestrian interactions?

A. Experiment Setup

Motion Primitives. In this work, we consider six instruction specifications representing common navigation motion primitives, summarized in Tab. I. Each motion primitive is associated with specific properties that a robot's trajectory must satisfy. To verify compliance with these properties, we develop a set of rule-based checks customized for each motion primitive.

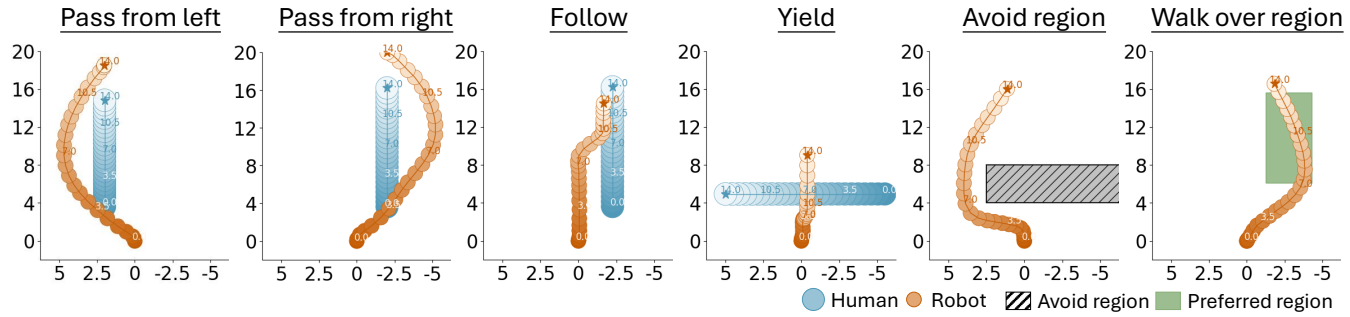
Simulation Environments. We evaluate ComposableNav in a 20m x 20m 2D simulation arena, with dynamic humans modeled as spheres and regions modeled as rectangles. For each instruction, 20 environments are randomly initialized, assigning initial positions and speeds to the entities based on the specific requirements of the instruction. The simulation operates with a control frequency of $\Delta t = 0.1$s, and each episode lasts a maximum of 300 timesteps, equivalent to 30.0 seconds.

ComposableNav Setup. Each diffusion model is trained for 25 diffusion steps using a cosine noise schedule to generate a fixed-length, time-dependent trajectory. The model is conditioned on either the observed human trajectory or a static region. The human trajectory consists of a sequence of time-dependent positions estimated under the constant velocity assumption, while the static region is represented as a rectangle defined by the positions of its four corners. Once the composed diffusion models generate a clean trajectory, we crop its endpoint to the nearest waypoint within a specified radius of the goal. We then implement Model Predictive Path Integral (MPPI) [14], a sampling-based MPC controller, to track the planned trajectory, and ComposableNav replans trajectories at a fixed interval to ensure continuous adaptation.

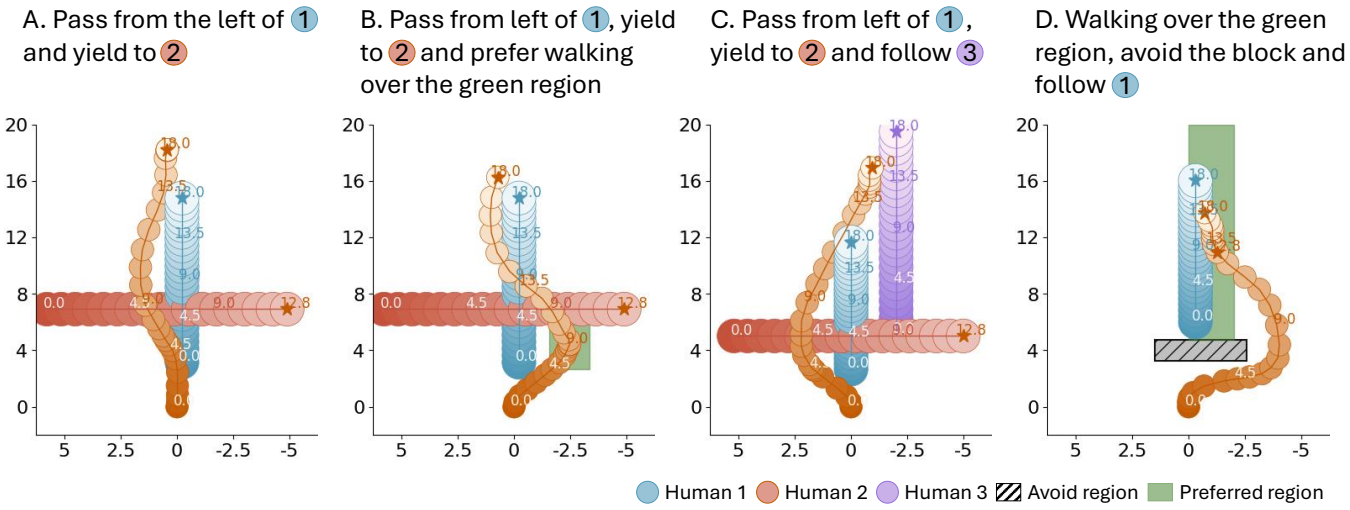
Metrics. We evaluate the success rate of all methods using four metrics: 1) Success Rate (SR), 2) Instruction Alignment (IA), 3) Free of Collision (CF), and 4) Goal Reaching (GR). Instruction Alignment (IA) checks if the executed trajectory satisfies all instruction specifications, where we use a rule-based method to identify whether the trajectory aligns with the instruction. A trajectory is considered successful (SR=1) if it meets all three criteria: following the instructions (IA=1), avoiding collisions (CF=1), and reaching the goal (GR=1).

B. Learning Motion Primitive

We first examine whether our proposed two-stage training procedure allows diffusion models to learn individual motion



(a) Motion Primitives



(b) Robot navigation behavior composed from motion primitives

Fig. 3: Simulation scenarios with different instruction specifications.

TABLE II: Performance of Pre-trained and Fine-tuned Diffusion Primitives in the Two-Step Training Procedure

MP	Pre-trained Model				Fine-tuned Model			
	SR(%)↑	IA(%)↑	CF(%)↑	GR(%)↑	SR(%)↑	IA(%)↑	CF(%)↑	GR(%)↑
L	44.0	44.0	100.0	100.0	100.0	100.0	100.0	100.0
R	37.0	37.0	100.0	100.0	100.0	100.0	100.0	100.0
F	27.0	27.0	100.0	100.0	99.0	100.0	100.0	99.0
Y	48.0	48.0	100.0	100.0	100.0	100.0	100.0	100.0
W	34.0	34.0	100.0	100.0	100.0	100.0	100.0	100.0
A	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

primitives without being explicitly trained on demonstration data. For each instruction specification listed in Tab. I, we create simulation environments containing a single entity with which the robot is instructed to interact. To evaluate the performance of ComposableNav, we test both pre-trained and fine-tuned diffusion models, as shown in Tab. II. Given the stochastic nature of diffusion models, we run each environment five times to reduce variance in the results. The results indicate that the pre-trained models consistently generate collision-free, goal-reaching trajectories, as they have been trained on datasets explicitly designed for this objective. Meanwhile, the fine-tuned models reliably follow instruction specifications without a significant decline in their ability to produce collision-free, goal-reaching trajectories. Note that the pre-trained model achieves a 100% success rate specifically for the *'Avoid walking through a region'* motion primitive, which is expected since avoiding designated regions shares the same objective as avoiding collisions with obstacles during the pre-training phase.

C. Composing Motion Primitives

TABLE III: Simulation Results

# MP	Method	SR(%)↑	IA(%)↑	CF(%)↑	GR(%)↑
2	VLM-Social-Nav [27]	6.9	6.9	95.6	100.0
	CoNVOI [9]	9.4	10.0	92.5	100.0
	BehAV [28]	16.9	17.5	**100.0**	90.6
	ComposableNav (ours)	**70.6**	**71.0**	96.9	**100.0**
3	VLM-Social-Nav [27]	0.0	0.6	96.9	91.9
	CoNVOI [9]	2.5	2.5	93.8	96.3
	BehAV [28]	15.0	17.5	**98.1**	86.9
	ComposableNav (ours)	**62.0**	**62.1**	97.6	**100.0**
4	VLM-Social-Nav [27]	0.6	1.3	81.9	82.5
	CoNVOI [9]	0.6	0.6	87.5	88.1
	BehAV [28]	6.3	10.6	**91.3**	70.0
	ComposableNav (ours)	**42.5**	**44.9**	86.0	**100.0**

To investigate whether ComposableNav can compose motion primitives to generate trajectories that follow previously

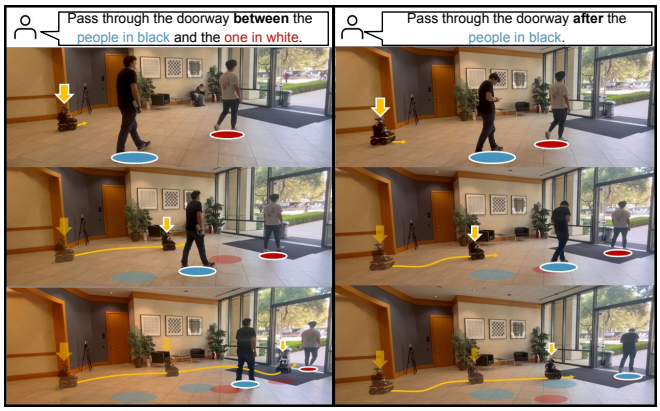


Fig. 4: Real world deployment.

unseen combinations of instruction specifications, we designed a testbed featuring 24 different specification combinations and quantitatively evaluated the performance of ComposableNav on this testbed, as shown in Tab. III. These combinations are categorized by complexity, ranging from two to four specifications. The complexity increases with the number of specifications, as the robot must account for more entities, leading to increasingly complex trajectory behaviors.

We compare ComposableNav with three VLM-based baseline methods in simulation: (1) VLM-Social-Nav [11], which leverages a VLM to select an action from predefined behaviors and translate it into a social cost function for planning with the Dynamic Window Approach (DWA) [29]; (2) CoNVOI [9], which uses a VLM to determine the robot's next waypoint from an annotated image and navigates to it using the DWA; and (3) BehAV [28], which employs a VLM to generate segmentation maps based on instructions and the navigation environment, converting them into cost maps for motion planning via a geometric planner. These baselines fall into two categories: the first two treat the VLM as a black-box policy that proposes a target action (e.g., next waypoint or velocity) for a geometric planner to track, while the third computes composable cost maps for planning.

We show the evaluation results in Tab. III and observe that ComposableNav outperforms all baselines in overall success rate by a significant margin. ComposableNav demonstrates superior instruction-following capabilities and consistently reaches the goal while maintaining comparable performance in collision avoidance. Additionally, we find that methods using VLMs as black-box policies generally perform poorly, as VLMs are not trained for such tasks and struggle to generate instruction-aligned navigation behaviors, especially as instructions become more complex. Finally, BehAV has the lowest goal-reaching rate, suggesting that methods relying on cost map composition often get stuck in local minima, resulting in unsuccessful navigation.

D. Deploy on Real Robot

We deployed ComposableNav on a Clearpath Jackal robot for real-world experiments. The robot is equipped with a Zed 2i camera for tracking human positions, which are used to estimate human trajectories as conditioning inputs for the diffusion models, and an Ouster LiDAR for obstacle detection, which the MPPI controller utilizes for collision avoidance. All computations run entirely onboard, leveraging an Intel i7-9700TE CPU and an NVIDIA RTX A2000 GPU.

To evaluate ComposableNav under real-world conditions, we replicated the experimental setup used in our simulation testbeds. For example, Fig. 4 illustrates scenarios where the robot attempts to navigate through a doorway but behaves differently depending on the instruction. Since failure cases in simulation—especially those caused by the diffusion model's difficulty in composing motion primitives—tend to also fail in the real world, we focused our evaluation on sim-to-real transferability by testing only scenarios that had succeeded in simulation. To this end, we conducted 40 controlled real-world trials, randomly selected from successful simulation runs, and observed a success rate of 35 out of 40, indicating strong transfer performance. Additionally, we profiled inference latency during deployment: the average inference time was 0.4 seconds during initial planning (when generating a plan from scratch), and just 0.06 seconds during replanning (when refining an existing plan), enabling real-time responsiveness.

We qualitatively analyzed failure cases of ComposableNav and identified two issues. First, human tracking errors occur when the person temporarily leaves the camera's field of view during turns, leading to misidentification despite a nearest-neighbor heuristic. Second, significant differences between replanned and previous paths cause the MPPI to generate abrupt acceleration or deceleration, resulting in jerky movements and overshooting. We hope to address these issues in future work.

VI. Related Work

A. Social Robot navigation

Social robot navigation focuses on enabling mobile robots to move smoothly through dynamic human environments while respecting social norms [27], [30]. These social norms generally include maintaining a comfortable distance from people to respect personal space, being polite by yielding to pedestrians, and following common rules like walking on the right side of the path [30]. Researchers have addressed this challenge using a variety of methods, including geometric rule-based approaches [31], [32], [33], [34], learning-based methods [35], [36], [37], [38], [39], and hybrid strategies that combine both [40], [41]. Recently, the use of vision-language models (VLMs) has gained traction in this field. One line of research leverages VLMs to directly propose robot actions, such as determining the next waypoints [9] or setting velocity constraints [11]. This approach takes advantage of the commonsense reasoning capabilities of VLMs to extract contextual information about the social environment from images [42] — information that is often challenging to obtain using traditional methods. However, VLMs currently face significant latency issues, making them unsuitable for the rapid decision-making required in dynamic social environments. Another line of research employs VLMs

to generate composable cost maps [28], which can be used with fast geometric planners to run in real time. While this approach addresses the latency issue, effectively composing cost maps to enable successful planning remains a challenge. In particular, ensuring the planner avoids local minima and generates a desirable trajectory can be difficult, especially when the optimal path requires navigating through regions that deviate significantly from typical, easily sampled routes.

B. Vision-Language Navigation/Action (VLN/VLA)

Vision-Language Navigation (VLN) [1], [2], [3] and Vision-Language Action (VLA) [43] integrate natural language understanding with visual perception to enable agents to navigate and perform tasks in 3D environments. VLN focuses on navigation guided by language instructions, while VLA extends to a broader range of tasks like object manipulation and interaction.

Early works introduced benchmarks and simulation environments [44], [45], [46], [47], [48], [49], [50] to train agents to navigate by grounding instructions in visual and spatial contexts [51], [52], [53], [54], [55], [56], [57], [58]. Recent approaches leveraging large vision-language models (VLMs) have advanced generalization beyond existing datasets, improving performance in novel and real-world settings [59], [60], [61], [62], [63], [64], [65]. Despite these advancements, both VLN and VLA overlook the social aspects of instructions and actions, leading to a significant limitation: the solutions developed by these methods fail to account for time sensitivity. In contrast, instruction-following solutions for social navigation must be time-critical to operate effectively in dynamic social human environments.

C. Diffusion for motion planning

Diffusion models [13], [12], [66] have gained popularity in robotics for motion planning tasks [67], [20]. Their ability to capture multimodal action distributions, handle high-dimensional outputs, and ensure stable training makes them well-suited for learning robot action policies [21]. Beyond these strengths, diffusion models offer the unique advantage of allowing their sampling process to be guided after training [68], [69]. In motion planning, this is achieved by framing the problem as planning-as-inference [24], [16], where classifier guidance [70] directs sampling from a posterior distribution to generalize actions beyond the training set. However, classifier-guidance requires a separate classifier, which can be challenging to obtain. Classifier-free guidance [15] addresses this limitation by incorporating conditional information directly into the diffusion model, removing the need for an external classifier. Potential-based Diffusion Motion Planning (PBDiff) [19] further demonstrates that diffusion models trained with classifier-free guidance can be composed together at inference time to generate new motion plans. Although diffusion models are trained on large dataset demonstrations, sampling specific behaviors — like consistently overtaking from the left or right — remains challenging. Denoising Diffusion Policy Optimization (DDPO) [17] addresses this by framing denoising as a multistep decision-making process, using policy gradient methods (e.g. PPO [18]) to fine-tune diffusion models for specific tasks. Inspired by PBDiff and DDPO, we pre-train a base diffusion model and finetune it with reinforcement learning objectives. Finally, we use classifier-free guidance to sample desired social navigation behaviors through a composition of finetuned diffusion models.

VII. Conclusion, Limitaitons, Future Works

We presented ComposableNav, a composable, diffusion-based motion planner for instruction-following navigation in dynamic environments. Unlike traditional methods, it decomposes instructions into specifications and composes motion primitives, simplifying training and improving generalization. ComposableNav has several limitations. First, the quality and diversity of pretraining data play a critical role in model generalization; our current approach relies on synthetic data generated in simulation, which may not fully capture the complexity of real-world environments. Second, the reward functions used during reinforcement learning finetuning are handcrafted and task-specific, limiting scalability to more abstract or ambiguous instructions. Third, our method assumes that instruction specifications and relevant entities can be accurately extracted by external perception and language models, which may not always hold in practice due to perception noise or ambiguous language. In future work, we aim to incorporate real-world navigation datasets to improve pretraining diversity, explore learning reward functions via vision-language models for greater scalability, and enhance robustness to perception errors through integrated uncertainty modeling and adaptive control.

References

[1] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. Wang, "Vision-and-language navigation: A survey of tasks, methods, and future directions," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7606–7623. [Online]. Available: https://aclanthology.org/2022.acl-long.524/

[2] Y. Zhang, Z. Ma, J. Li, Y. Qiao, Z. Wang, J. Chai, Q. Wu, M. Bansal, and P. Kordjamshidi, "Vision-and-language navigation today and tomorrow: A survey in the era of foundation models," *Transactions on Machine Learning Research*, 2024, survey Certification. [Online]. Available: https://openreview.net/forum?id=yiqeh2ZYUh

[3] S.-M. Park and Y.-G. Kim, "Visual language navigation: a survey and open challenges," *Artif. Intell. Rev.*, vol. 56, no. 1, p. 365–427, Mar. 2022. [Online]. Available: https://doi.org/10.1007/s10462-022-10174-9

[4] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 2165–2183. [Online]. Available: https://proceedings.mlr.press/v229/zitkovich23a.html

[5] B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: a comprehensive survey," *Artif. Intell. Rev.*, vol. 55, no. 2, p. 945–990, Feb. 2022. [Online]. Available: https://doi.org/10.1007/s10462-021-09997-9

[6] J. Hu, R. Hendrix, A. Farhadi, A. Kembhavi, R. Martin-Martin, P. Stone, K.-H. Zeng, and K. Ehsani, "Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning," 2024. [Online]. Available: https://arxiv.org/abs/2409.16578

[7] W. Liu, N. Nie, R. Zhang, J. Mao, and J. Wu, "Learning compositional behaviors from demonstration and language," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=fR1rCXjCQX

[8] V. Myers, C. Zheng, O. Mees, K. Fang, and S. Levine, "Policy adaptation via language optimization: Decomposing tasks for few-shot imitation," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=qUSa3F79am

[9] A. J. Sathyamoorthy, K. Weerakoon, M. Elnoor, A. Zore, B. Ichter, F. Xia, J. Tan, W. Yu, and D. Manocha, "Convoi: Context-aware navigation using vision language models in outdoor and indoor environments," 2024. [Online]. Available: https://arxiv.org/abs/2403.15637

[10] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, Q. Vuong, T. Zhang, T.-W. E. Lee, K.-H. Lee, P. Xu, S. Kirmani, Y. Zhu, A. Zeng, K. Hausman, N. Heess, C. Finn, S. Levine, and B. Ichter, "Pivot: Iterative visual prompting elicits actionable knowledge for vlms," 2024.

[11] D. Song, J. Liang, A. Payandeh, A. H. Raj, X. Xiao, and D. Manocha, "Vlm-social-nav: Socially aware robot navigation through scoring using vision-language models," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 508–515, 2025.

[12] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[13] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2256–2265. [Online]. Available: https://proceedings.mlr.press/v37/sohl-dickstein15.html

[14] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1433–1440.

[15] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. [Online]. Available: https://openreview.net/forum?id=qw8AKxfYbI

[16] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[17] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine, "Training diffusion models with reinforcement learning," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=YCWjhGrJFD

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[19] Y. Luo, C. Sun, J. B. Tenenbaum, and Y. Du, "Potential based diffusion motion planning," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=Qb68Rs0p9f

[20] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[21] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[22] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially CompliAnt Navigation Dataset (SCAND): A Large-Scale Dataset of Demonstrations for Social Navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 807–11 814, 2022.

[23] A. Zhang, C. Eranki, C. Zhang, J.-H. Park, R. Hong, P. Kalyani, L. Kalyanaraman, A. Gamare, A. Bagad, M. Esteva, and J. Biswas, "Toward robust robot 3-d perception in urban environments: The ut campus object dataset," *IEEE Transactions on Robotics*, vol. 40, pp. 3322–3340, 2024.

[24] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1916–1923, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260191316

[25] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum, "Compositional visual generation with composable diffusion models," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 423–439.

[26] S. Zhou, Y. Du, S. Zhang, M. Xu, Y. Shen, W. Xiao, D.-Y. Yeung, and C. Gan, "Adaptive online replanning with diffusion models," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2023.

[27] *J. Hum.-Robot Interact.*, vol. 12, no. 3, 2023.

[28] K. Weerakoon, M. Elnoor, G. Seneviratne, V. Rajagopal, S. H. Arul, J. Liang, M. K. M. Jaffar, and D. Manocha, "Behav: Behavioral rule guided autonomy using vlms for robot navigation in outdoor scenes," 2024. [Online]. Available: https://arxiv.org/abs/2409.16484

[29] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," Tech. Rep., 1995.

[30] A. Francis, C. Pérez-D'arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra, H.-T. L. Chiang, M. Everett, S. Ha, J. Hart, J. P. How, H. Karnan, T.-W. E. Lee, L. J. Manso, R. Mirksy, S. Pirk, P. T. Singamaneni, P. Stone, A. V. Taylor, P. Trautman, N. Tsoi, M. Vázquez, X. Xiao, P. Xu, N. Yokoyama, A. Toshev, and R. Martín-Martín, "Principles and guidelines for evaluating social robot navigation algorithms," *J. Hum.-Robot Interact.*, 2024, just Accepted. [Online]. Available: https://doi.org/10.1145/3700599

[31] J. Holtz, S. Andrews, A. Guha, and J. Biswas, "Iterative program synthesis for adaptable social navigation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6256–6261.

[32] B. Holman, A. Anwar, A. Singh, M. Tec, J. Hart, and P. Stone, "Watch where you're going! gaze and head orientation as predictors for social robot navigation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3553–3559.

[33] A. Wang, C. Mavrogiannis, and A. Steinfeld, "Group-based motion prediction for navigation in crowded environments," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=knObbYqSowX

[34] M. Kollmitz, K. Hsiao, J. Gaa, and W. Burgard, "Time dependent planning on a layered social cost map for human-aware robot navigation," in *2015 European Conference on Mobile Robots (ECMR)*, 2015, pp. 1–6.

[35] Z. Xie, P. Xin, and P. Dames, "Towards safe navigation through crowded dynamic environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4934–4940.

[36] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3517–3524.

[37] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 015–12 021.

[38] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1343–1350.

[39] S. Liu, H. Xia, F. C. Pouria, K. Hong, N. Chakraborty, and K. Driggs-Campbell, "Height: Heterogeneous interaction graph transformer for robot navigation in crowded and constrained environments," 2024. [Online]. Available: https://arxiv.org/abs/2411.12150

[40] Z. Zheng, C. Cao, and J. Pan, "A hierarchical approach for mobile

robot exploration in pedestrian crowd," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 175–182, 2022.

[41] A. H. Raj, Z. Hu, H. Karnan, R. Chandra, A. Payandeh, L. Mao, P. Stone, J. Biswas, and X. Xiao, "Rethinking social robot navigation: Leveraging the best of two worlds," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 16 330–16 337.

[42] D. Song, J. Liang, A. Payandeh, A. H. Raj, X. Xiao, and D. Manocha, "Vlm-social-nav: Socially aware robot navigation through scoring using vision-language models," 2024. [Online]. Available: https://arxiv.org/abs/2404.00210

[43] Y. Ma, Z. Song, Y. Zhuang, J. Hao, and I. King, "A survey on vision-language-action models for embodied ai," 2024. [Online]. Available: https://arxiv.org/abs/2405.14093

[44] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:4673790

[45] J. Krantz, E. Wijmans, A. Majundar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision and language navigation in continuous environments," in *European Conference on Computer Vision (ECCV)*, 2020.

[46] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 4392–4412.

[47] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[48] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 530–12 539.

[49] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "AI2-THOR: An Interactive 3D Environment for Visual AI," *arXiv*, 2017.

[50] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian, "Building generalizable agents with a realistic and rich 3d environment," *arXiv preprint arXiv:1801.02209*, 2018.

[51] S. Raychaudhuri, S. Wani, S. Patel, U. Jain, and A. Chang, "Language-aligned waypoint (LAW) supervision for vision-and-language navigation in continuous environments," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4018–4028. [Online]. Available: https://aclanthology.org/2021.emnlp-main.328/

[52] M. Z. Irshad, N. Chowdhury Mithun, Z. Seymour, H.-P. Chiu, S. Samarasekera, and R. Kumar, "Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments," in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022, pp. 4065–4071.

[53] K. Chen, J. K. Chen, J. Chuang, M. Vázquez, and S. Savarese, "Topological planning with transformers for vision-and-language navigation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 271–11 281.

[54] G. Georgakis, K. Schmeckpeper, K. Wanchoo, S. Dan, E. Miltsakaki, D. Roth, and K. Daniilidis, "Cross-modal map learning for vision and language navigation," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15 439–15 449, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247362748

[55] Y. Hong, Z. Wang, Q. Wu, and S. Gould, "Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 418–15 428.

[56] J. Krantz, A. Gokaslan, D. Batra, S. Lee, and O. Maksymets, "Waypoint models for instruction-guided navigation in continuous environments," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15 142–15 151, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238354099

[57] S. Liu, A. Hasan, K. Hong, R. Wang, P. Chang, Z. Mizrachi, J. Lin, D. L. McPherson, W. A. Rogers, and K. Driggs-Campbell, "Dragon: A dialogue-based robot for assistive navigation with visual language grounding," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3712–3719, 2024.

[58] P. Chang, S. Liu, and K. Driggs-Campbell, "Learning visual-audio representations for voice-controlled robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[59] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *Robotics: Science and Systems*, 2024.

[60] D. Shah, B. Osinski, B. Ichter, and S. Levine, "LM-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=UW5A3SweAH

[61] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[62] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as i can and not as i say: Grounding language in robotic affordances," in *arXiv preprint arXiv:2204.01691*, 2022.

[63] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: an embodied multimodal language model," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[64] Z. Xu, H.-T. L. Chiang, Z. Fu, M. G. Jacob, T. Zhang, T.-W. E. Lee, W. Yu, C. Schenck, D. Rendleman, D. Shah, F. Xia, J. Hsu, J. Hoech, P. Florence, S. Kirmani, S. Singh, V. Sindhwani, C. Parada, C. Finn, P. Xu, S. Levine, and J. Tan, "Mobility VLA: Multimodal instruction navigation with long-context VLMs and topological graphs," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=JScswMfEQ0

[65] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "Instructnav: Zero-shot system for generic instruction navigation in unexplored environment," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=fCDOfpTCzZ

[66] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," 2021. [Online]. Available: https://arxiv.org/abs/2102.09672

[67] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *Arxiv*, 2024.

[68] Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. Grathwohl, "Reduce, reuse, recycle: compositional generation with energy-based diffusion models and mcmc," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[69] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=PxTIG12RRHS

[70] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., 2024.