Hierarchical Cross-Modal Agent for Robotics Vision-and-Language Navigation

Muhammad Zubair Irshad*, Chih-Yao Ma*†, Zsolt Kira*

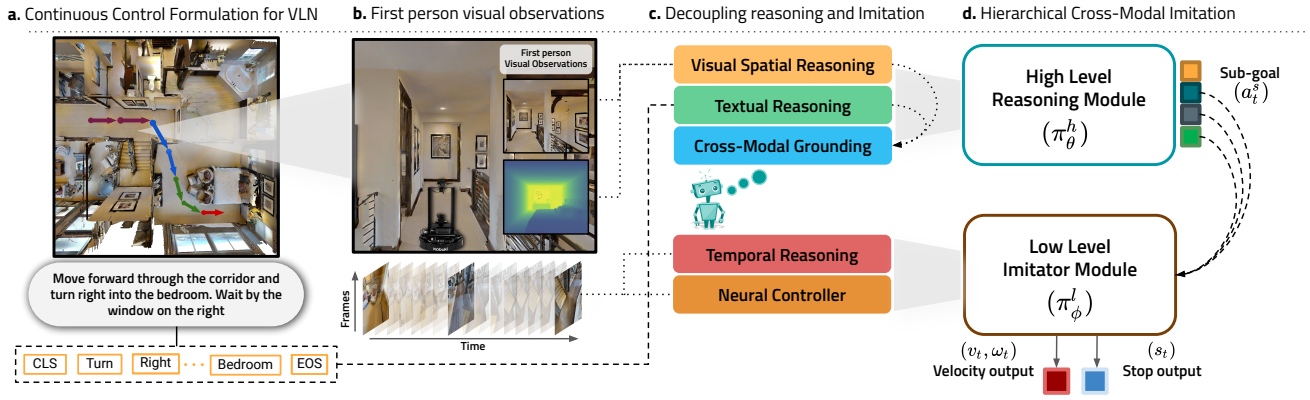

Fig. 1: **Overview:** Robotics Vision-and-Language Navigation (Robo-VLN) task in continuous environments and our proposed Hierarchical Cross-Modal (HCM) agent. The agent decouples reasoning and imitation through a modularized training regime to solve the complex long-horizon Robo-VLN task.

***Abstract*— Deep Learning has revolutionized our ability to solve complex problems such as Vision-and-Language Navigation (VLN). This task requires the agent to navigate to a goal purely based on visual sensory inputs given natural language instructions. However, prior works formulate the problem as a navigation graph with a discrete action space. In this work, we lift the agent off the navigation graph and propose a more complex VLN setting in continuous 3D reconstructed environments. Our proposed setting, Robo-VLN, more closely mimics the challenges of real world navigation. Robo-VLN tasks have longer trajectory lengths, continuous action spaces, and challenges such as obstacles. We provide a suite of baselines inspired by state-of-the-art works in discrete VLN and show that they are less effective at this task. We further propose that *decomposing the task* into specialized high- and low-level policies can more effectively tackle this task. With extensive experiments, we show that by using layered decision making, modularized training, and decoupling reasoning and imitation, our proposed Hierarchical Cross-Modal (HCM) agent outperforms existing baselines in all key metrics and sets a new benchmark for Robo-VLN.**

I. Introduction

The promise of personal assistant robots that can seamlessly follow human instructions in real life environments has long been sought after. Recent advancements in deep learning (to extract meaningful information from raw sensor data) and deep reinforcement learning (to learn effective decision-making policies) have enabled some progress towards this goal [1, 2, 3]. Due to the difficulty of collecting data in these contexts, a great deal of work has been done using photo-realistic simulations such as those captured through Matterport3D panoramas in homes [4] or point-cloud meshes in Gibson [5]. For example, a number of works have investigated autonomous agents that can follow rich, natural-language instructions in such simulations [6, 7, 8]. Precisely defined, Vision-and-Language Navigation (VLN) is a task which requires the agent to navigate to a goal location purely based on visual inputs and provided instructions in the absence of a prior global map [9].

While increasingly effective neural network architectures have been developed for these tasks, many limitations still exist that prevent their applicability to real-world robotics problems. Specifically, previous works [4, 7, 8, 10, 11] have focused on a simpler subset of this problem by defining the instruction-guided robot trajectories as either a discrete navigation graph [4, 9] or assuming the action space of the autonomous agent comprises of discrete values [12, 13]. These formulations assume known topology, perfect localization and deterministic navigation from one viewpoint to the next in the absence of any obstacles [13]. Hence these assumptions significantly deviate from the real world both in terms of control and perception.

As a first contribution, we focus on a richer VLN formulation which is defined in continuous environments over long horizon trajectories. Our proposed setting, **Robo-VLN** (**Robo**tics **V**ision-and-**L**anguage **N**avigation), is summarized in Figure 1 and Section III. We lift the agent off the navigation graph, making the language guided navigation problem richer, more challenging, and closer to the real world.

In an attempt to solve the language-guided navigation (VLN) problem, recent learning-based approaches [6, 16, 17] make use of sequence-to-sequence architectures [18]. However, when tested for generalization performance in un-

*Georgia Institute of Technology (mirshad7,zkira)@gatech.edu
†Now at Facebook cyma@fb.com

TABLE I: **Comparison between our proposed Robo-VLN setting and prior environments used for Vision-and-Language Navigation**

	—Simulation—		—Environment—			—Instructions—	
	Action space	**Granularity**	**Agent**	**Navigation**	**Type**	**Richness**	**Generation**
Touchdown [14], R2R [9]	Discrete	High	Virtual	Unconstrained	Photo-realistic	Complex	Human-annotated
Follow-net [10]	Discrete	High	Virtual	Constrained	Synthetic	Simple	Human-annotated
LANI [15]	Discrete	High	Virtual	Constrained	Synthetic	Simple	Template based
VLN-CE [13]	Discrete	High	Virtual	Unconstrained	Photo-realistic	Complex	Human-annotated
Robo-VLN (Ours)	Continuous	High/Low	Robotics	Unconstrained	Photo-realistic	Complex	Human-annotated

seen environments, these approaches (initially developed for shorter horizon nav-graph problems) translate poorly to more complex settings [12, 13], as we also showed for Robo-VLN in Section VI. Hence, for our proposed continuous VLN setting over long-horizon trajectories, we present an approach utilizing *hierarchical decomposition*. Our proposed method leverages hierarchy to decouple cross-modal reasoning and imitation, thus equipping the agent with the following key abilities:

1. Decouple Reasoning and Imitation. The agent is comprised of a high-level policy and a corresponding low-level policy. The high-level policy is tasked with aligning the relevant instructions with observed visual cues as well as reasoning over which instructions have been completed, hence producing a sub-goal output through cross-modal grounding. The low-level policy imitates the feedback controller based on sub-goal information and observed visual states. A layered decision making allows spatially different reasoning at different levels in the hierarchy, hence specializing each policy with a dedicated reasoning abstraction level.

2. Modularized Training. Disentangling reasoning and controls allows fragmenting a complex long horizon problem into shorter time horizon problems. Since each policy is tasked with fulfilling a dedicated goal, each module utilizes separate end-to-end training with sparse communication between the hierarchy in terms of sub-goal information. In summary, we make the following contributions:

- To the best of our knowledge, we present the first work on formulating Vision-and-Language Navigation (VLN) as a continuous control problem in photo-realistic simulations, hence lifting the agent of the assumptions enforced by navigation graphs and discrete action spaces.
- We formulate a novel hierarchical framework for Robo-VLN, referred to as **H**ierarchical **C**ross-**M**odal Agent (**HCM**) for effective attention between different input modalities through a modularized training regime, hence tackling a long-horizon and cross-modal task using layered decision making.
- Provide a suite of baseline models in Robo-VLN inspired by recent state-of-the-art works in VLN and present a comprehensive comparison against our proposed hierarchical approach — Our work sets a new strong benchmark performance for a long horizon complex task, Robo-VLN, with over 13% improvement in absolute success rate in unseen validation environments.

II. Related Work

Vision-and-Language Navigation. Learning based navigation has been explored in both synthetic [19, 20, 21] and photo-realistic [4, 5, 22] environments. For a navigation graph based formulation of the VLN problem (i.e. discrete action space), previous works have utilized hybrid reinforcement learning [23], behavior cloning [24], speaker-follower [25] and sequence to sequence based approaches [9]. Subsequent methods have focused on utilizing auxiliary losses [16, 26], backtracking [6] and cross-modal attention techniques [27, 28, 29] to improve the performance of VLN agents. Our work, in contrast to discrete VLN setting [9, 13] (see Table I), focuses on a much richer VLN formulation, which is defined for continuous action spaces over long-horizon trajectories. We study the new continuous Robo-VLN setting and propose hierarchical cross-modal attention and modularized training regime for such task.

Hierarchical Decomposition. Hierarchical structure is most commonly utilized in the context of Reinforcement Learning over long-time horizons to improve sample efficiency [30, 31, 32]. Our work closely relates to the options framework in Reinforcement Learning [24, 30, 33, 34] where the top-level policy identifies high-level decisions to be fulfilled by a bottom-level policy. In relation to other works which utilize sub-task decomposition for behaviour cloning [33, 35], we show that decomposing hierarchy based on reasoning and imitation are quite effective for long-horizon multi-modal tasks such as Robo-VLN.

III. Robotics Vision-and-Language Navigation Environment (Robo-VLN)

Different from existing VLN environments, we propose a new continuous environment for VLN that more closely mirrors the challenges of the real world, Robo-VLN — a continuous control formulation for Vision-and-Language Navigation. Compared to navigation graph based [9] and discrete VLN settings [13], Robo-VLN provides longer horizon trajectories (4.5x average number of steps), more visual frames (∼3.5M visual frames), and a balanced high-level action distribution (see Figure 2). Hence, making the problem more challenging and closer to the real-world.

A. Problem Definition

Formally, consider an autonomous agent $\tilde{\mathcal{A}}$ in an unknown environment $\tilde{\mathcal{E}}$. The goal of a Robo-VLN agent is to learn a policy $a_t = \pi(x_t, q_t, \theta)$ where the agent receives visual observations (x_t) from the environment $\tilde{\mathcal{E}}$ at each time-step

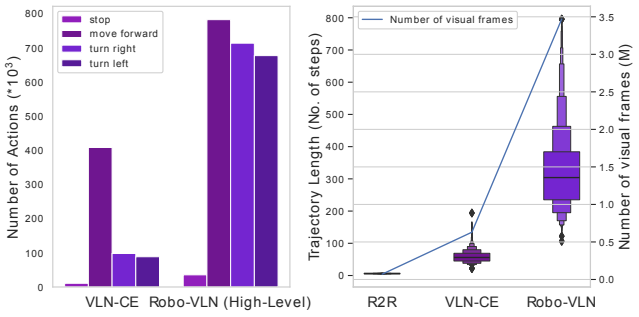

Fig. 2: **Robo-VLN compared with discrete VLN settings:** VLN-CE [13] and R2R [9]. We provide longer horizon trajectories (4.5x average number of steps, over 3M visual frames, and a balanced high-level action distribution.

(t) while following a provided instruction (q) to navigate to a goal location $\mathcal{G}$. θ denotes the learnable parameters of the policy π. The action space of the agent consists of continuous linear and angular velocity (v_t, ω_t) and a discrete stop action (s_t). An episode (τ) is considered successful if agent's distance to the goal is less than a threshold $(d_a < 3m)$ and the agent comes to a stop by either taking the stop action (s_t) or decreasing its angular velocity below a certain threshold.

B. Constructing Continuous VLN in 3D Reconstructions

To make the continuous VLN formulation possible in 3D reconstructed environments, we port over human annotated instructions (q_t) corresponding to sparse way-points (z_t) along each instruction-trajectory pair in Room2Room (R2R) dataset [9], using a continuous control formulation. We do this in 2 stages as follows:

Ground-truth oracle feedback controller in 3D reconstructed environments. We consider the robotic agent to be a differential drive mobile robot, Locobot [36], with a specified radius and height. We develop $A^\star$ planner to compute high-level oracle actions (a_t^h) along the shortest path to the goal and use a feedback controller [37] to convert the discrete R2R trajectories [9] into continuous ones. The low-level oracle controller (u_t) outputs velocity commands (v_t, ω_t) given sparse way-points (z_t) along a given language-guided navigation trajectory from the R2R dataset [4]. The converted continuous actions from the low-level controller will then be used as ground-truth low-level supervisions a_t^l when training the navigation agents. We create this continuous control formulation inside Matterport 3D environments [4] by considering the Locobot robot as a 3D mesh inside 3D reconstructed environments (see Figure 5). We use the robot's dynamics [38] to predict next state $(\hat{x}_{t+1})$ given current state $(\hat{x}_t)$ and controller actions (a_t^l). Similar to Habitat [22], we render the mesh for any arbitrary viewpoint by taking the position generated by the dynamic model inside the 3D reconstruction.

Obtaining Navigable Instruction-Trajectory pairs. Given a feedback controller of the form $a_t^l = u_t(z_t)$ and high-level sparse viewpoints $(z_t = [z_1, \ldots, z_N]$ along the language guided navigation trajectory inside a reconstructed mesh, we search for the navigable space $h_{nav}(z_t)$ using collision detection. We find navigable space for all the trajectories present in the R2R dataset [9]. This procedure ensures the transfer of only the navigable trajectories from R2R dataset to the continuous control formulation in Robo-VLN; hence, we eliminate non-navigable unrealistic paths for a mobile robot, such as climbing up the stairs and moving through obstacles. Through this approach, we transferred 71% of the trajectories from the discrete VLN setting (VLN-CE [13]) while preserving all the environments in the Matterport3D dataset [4]. At the end, Robo-VLN's expert demonstration provide first person RGB-D visual observations (i_t), human instructions (q_t), and oracle actions (a_t^h, a_t^l) for each instruction-trajectory pair.

IV. Hierarchical Cross-Modal Agent

Learning an effective policy (π) for a long horizon continuous control problem entails preserving the temporal states as well as spatially reasoning about the surroundings. We therefore propose a hierarchical agent to tackle the Robo-VLN task as it effectively disentangles different dedicated tasks through layered decision making. Given states $(\mathcal{X} = \{x\})$ and instructions $(\mathcal{Q} = \{q\})$, our agent leverages these inputs and learns a high-level policy $(\pi_\theta^h : \mathcal{X} \times \mathcal{Q} \rightarrow \mathcal{A}_{s,t})$ and a corresponding low-level policy $(\pi_\gamma^l : \mathcal{X} \times \mathcal{A}_{s,t} \rightarrow \mathcal{A}_{l,t})$. The high-level policy consistently reasons about the alignment between input textual and visual modalities to produce a sub-goal output $(\mathcal{A}_{s,t})$. The low-level policy ensures that the high-level sub-goal is translated to low-level actions $(\mathcal{A}_{l,t})$ effectively by imitating the expert controller through an imitation learning policy. Our approach is summarized in Figure 3 and subsequent sections.

A. High-Level Policy

The high-level policy (π_θ^h) decides a short-term goal (a_t^h) based on the input instructions (q_t) and observed visual information $x_t = \{r_t, d_t\}$ from the environment at each time-step, where r_t, d_t denote the RGB and Depth sensor readings respectively. π_θ^h consists of an encoder-decoder architecture with cross attention between the modules. Subsequent modules of the high-level policy (π_θ^h) are described below.

Multi-Modal Cross Attention Encoder. Given a natural language instruction comprised of k words, we denote its feature representation as $\{q = q_t^1, q_t^2, \ldots, q_t^k\}$, where q_t^i is the encoded feature representation of the i_{th} word using BERT embedding [39] to extract meaningful representation of words in the sentence. To encode the observed RGB-D states $(r_t \in \mathbb{R}^{h_o \times w_o \times 3}$, $d_t \in \mathbb{R}^{h_o \times w_o})$, we generate a low-resolution spatial feature representations $f_r \in \mathbb{R}^{H_s \times W_s \times C_s}$ and $f_d \in \mathbb{R}^{H_s \times W_s \times C_s}$ by using a pre-trained ConvNet backbone, where $H_s = W_s = 7$ and $C_s = 2048$. At each time-step t, we combine the individual RGB (f_r) and Depth (f_d) spatial features with encoded language representation (q_t) using a Transformer module [40]. Each Transformer module is comprised of stacked multi-head attention block $(\mathcal{A}_M)$ followed by a position-wise feed-forward block. We utilize layer normalizations [41] between these blocks along

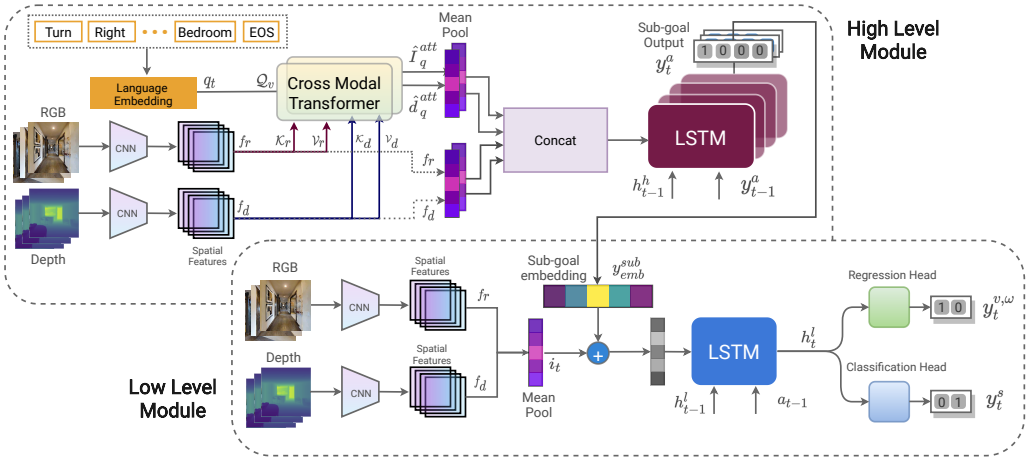

Fig. 3: **Hierarchical Cross-Modal Agent (HCM):** Our proposed agent consists of a *high-level module* and a corresponding *low-level module*. High-level module predicts the sub-goal output based on alignment between instructions and visual observations. Low-level module translates the high-level sub-goal output to linear and angular velocities using an imitation learning policy.

with the residual connection from the previous block such that output of each individual block is $LayerNorm(z + module(z))$. Each Transformer block is computed as follows:

$$\begin{aligned} \mathcal{A}_M(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) &= \text{concat}(\boldsymbol{h}_1,\ldots,\boldsymbol{h}_k)\boldsymbol{W}^h, \\ \text{where } \boldsymbol{h}_i &= \mathcal{A}\left(\boldsymbol{Q}W_i^Q, \boldsymbol{K}W_i^K, \boldsymbol{V}W_i^V\right) \\ \mathcal{A}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) &= \text{softmax}\left(\frac{\boldsymbol{Q}K^T}{\sqrt{d_k}}\right)\boldsymbol{V} \end{aligned} \tag{1}$$

The Attention output ($\mathcal{A}$) is a weighted sum of the values (V) calculated using a similarity between projected Query (Q) and Key (K). $\mathcal{A}_M$ represents stacked Attention blocks ($\mathcal{A}$), and W_i^Q, W_i^K, W_i^V and W^h are parameters to be learnt.

We utilize Equation 1 to perform cross attention between visual spatial representation (RGB f_r or Depth f_d) and language features (q_t) successively. We do this by utilising the sum of language features and sinusoidal Positional Encoding (PE [40]) as query ($Q = q_t$ + PE($\boldsymbol{q}_t$)) and visual representation as Key ($K_r = \boldsymbol{f}_r$ or $K_d = \boldsymbol{f}_d$) as well as Value ($V_r = \boldsymbol{f}_r$ or $V_d = \boldsymbol{f}_d$). The final outputs, which we denote as cross-attended context (from RGB or Depth), are computed using $\mathcal{A}_M(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})$, *e.g.*, $\hat{\boldsymbol{I}}_q^{att}$ for RGB input and $\hat{\boldsymbol{d}}_q^{att}$ for Depth input.

These cross-attended contexts represent the matching between instructions and corresponding visual features at each time step t. Note that the learnable weights in the Transformer are not shared between the two modalities.

Multi-Modal Attention Decoder. To decide on which direction to go next and select the most optimal high-level action (a_t^h) high-level policy preserves a temporal memory of the attended visual-linguistic contexts ($\hat{\boldsymbol{I}}_q^{att}$, $\hat{\boldsymbol{d}}_q^{att}$), mean-pooled visual features ($\hat{\mathrm{v}}_t$) and previous actions (a_{t-1}^h). We rely on a Recurrent Neural Network to preserve this temporal information across time.

$$\begin{aligned} \boldsymbol{h}_t^h &= \text{LSTM}\left(\left[\hat{\boldsymbol{I}}_q^{att}, \hat{\boldsymbol{d}}_q^{att}, \hat{\mathbf{v}}_t, a_{t-1}, \boldsymbol{h}_{t-1}^h\right]\right) \\ \hat{\mathbf{v}}_t &= \boldsymbol{W}_i(\mathrm{g}([\boldsymbol{f}_r, \boldsymbol{f}_d])^\top + \boldsymbol{b}_i) \end{aligned} \tag{2}$$

where $\mathrm{g}(.)$ is mean adaptive pooling across the spatial dimensions. W_i and b_i are learned parameters of a fully-connected layer.

The agent computes a probability (p_a^h) of selecting the most optimal action (a_t) at each time-step by employing a feed-forward network followed by a $softmax$ as follows:

$$p_a^h = softmax(\boldsymbol{W}_a([\boldsymbol{h}_t^h] + \boldsymbol{b}_a)) \tag{3}$$

where W_a and b_a are parameters to be learnt. High-level action a_t comprises of the following navigable directions: `move forward` (0.25m), `turn-left` or `turn-right` (15 degrees) and `stop`.

B. Low-level Policy

We employ an imitation policy for the low-level module. At each time-step t, the low-level policy (π_ϕ^l) selects a low-level action (a_t^l)) given the sub-goal (a_t^h), generated by the high-level policy and observed visual states (r_t, d_t) from the environment. Low-level actions are comprised of agent's linear and angular velocity (v_t, ω_t). Similar to the high-level module, we use mean pooled visual features ($\hat{\mathrm{v}}_t$) for the low-level policy and additionally condition the policy on the high-level sub-goal (a_t^h). Furthermore, we utilize stacked LSTM layers with respective fully-connected layers to generate both low-level action and stop probabilities (p_a^l, p_a^s):

$$\boldsymbol{h}_t^l = \text{LSTM}\left(\left[\hat{\boldsymbol{v}}_t, \boldsymbol{a}_t^h, \boldsymbol{h}_{t-1}^l\right]\right) \tag{4}$$

$$p_a^h = tanh(\mathrm{g_a}([\boldsymbol{h}_t^l, \boldsymbol{a}_{t-1}^l])), \quad p_a^s = \sigma(\mathrm{g_s}([\boldsymbol{h}_t^l, \boldsymbol{a}_{t-1}^l])) \tag{5}$$

where $\mathrm{g_a}(.)$ and $\mathrm{g_s}(.)$ are one-layer Multi-Layer Perceptrons (MLP). σ and $tanh$ are sigmoid and tanh activation functions respectively.

C. Training Details

We train both high- and low-level policies jointly with three different losses. We employ a multi-class cross-entropy loss computed between ground-truth high-level navigable action (y_t^a) and the predicted action probability (p_a^h) for

TABLE II: **Quantitative comparison**: Comparison with strong baselines. Note that these baselines are reimplementations from VLN-CE [13] with small changes (see Section VI for further details).

	Method	Validation Seen					Validation Unseen				
		SR ↑	**SPL** ↑	**NDTW** ↑	**TL** ↑	**NE** ↓	**SR** ↑	**SPL** ↑	**NDTW** ↑	**TL** ↑	**NE** ↓
1	Random Agent	0.07	0.07	0.14	5.26	10.25	0.08	0.08	0.14	5.40	9.81
2	Seq2Seq [4]	0.36	0.34	0.32	11.84	8.63	0.33	0.30	0.28	11.92	8.97
3	PM [16]	0.32	0.27	0.23	14.12	9.33	0.28	0.24	0.22	13.85	9.82
4	CMA [17]	0.28	0.25	0.22	11.52	9.95	0.28	0.25	0.23	11.57	9.63
	HCM (Ours)	**0.49**	**0.43**	**0.35**	**13.53**	**7.48**	**0.46**	**0.40**	**0.35**	**14.06**	**7.94**

TABLE III: **Ablation Study**: Impact of different modules and design choices in our proposed Hierarchical agent.

	#	Module			Validation Seen					Validation Unseen				
		Vision	Hierarchy	RGB-D Early fusion	**SR** ↑	**SPL** ↑	**NDTW** ↑	**TL** ↑	**NE** ↓↑	**SR** ↑	**SPL**	**NDTW** ↑	**TL** ↑	**NE** ↓
Hierarchical Agent	1		✓		0.07	0.07	0.14	4.82	10.34	0.07	0.07	0.14	10.2	4.81
	2	✓			0.44	0.37	0.31	14.87	8.21	0.40	0.34	0.28	15.32	8.64
	3	✓	✓	✓	0.39	0.35	0.29	13.87	9.13	0.34	0.31	0.28	12.85	8.78
	4	✓	✓		0.49	0.43	0.35	13.53	7.48	0.46	0.40	0.35	14.06	7.94

the high-level policy. We employ a mean squared error loss between ground-truth velocity commands ($y_t^{v,\omega}$) and predicted low-level action probabilities (p_a^l). Lastly, we use a binary cross-entropy loss between ground-truth stopping actions (y_t^s) and predicted stop probabilities (p_a^s) as follows:

$$\mathcal{L}_{\text{loss}} = \lambda \overbrace{\sum_{t=1}^{T} y_t^a \log\left(p_h^a\right)}^{\text{High-Level Action Loss}} + (1-\lambda)\Big(\overbrace{\sum_{t=1}^{T}\left(y_t^{v,\omega} - p_a^l\right)^2}^{\text{Low-Level Action Loss}} + \overbrace{\sum_{t=1}^{T} y_t^s \log\left(p_a^s\right)}^{\text{Low-Level Stop Loss}}\Big)$$

V. Dataset and Implementations

Simulation and Dataset. We use Habitat simulator [22] to perform our experiments. Our dataset, Robo-VLN, is built upon Matterport3D dataset [4], which is a collection of 90 environments captured through around 10k high-definition RGB-D panoramas. Robo-VLN provides 3177 trajectories, and each trajectory is associated with 3 instructions annotated by humans ported over from the R2R Dataset [9]. Overall, the dataset comprises 9533 expert instruction-trajectory pairs with an average trajectory length of 326 steps compared to 55.8 in VLN-CE [13] and 5 in R2R [9]. The corresponding dataset is divided into train, validation seen and validation unseen splits.

Evaluation Metrics. We evaluate our experiments on the following key standard metrics described by Anderson et al. [42] and Gabriel et al. [43]: Success rate (**SR**), Success weighted by path length (**SPL**), Normalized Dynamic Time Warping (**NDTW**), Trajectory Length (**TL**) and Navigation Error (**NE**). We use SPL and NDTW as the primary metrics for comparison. Both of these metrics measure the deviation from ground-truth trajectories; SPL places more emphasis on reaching the goal location, whereas NDTW emphasises on following the complete ground-truth path.

Implementation Details. We use pre-trained ResNet-50 on ImageNet [44] and pre-trained ConvNet on a large scale point-goal navigation task, DDPPO [45] to extract spatial features for images and depth modalities successively. For transformer module, we use a hidden size ($H = 256$), number of Transformer heads ($n_h = 4$), and the size of feed-forward layer ($FF = 1024$). We found that truncated backpropagation through time [46] was invaluable to train longer sequence recurrent networks in our case. We used a truncation length of 100 to train attention decoders in both policies. We trained the network for 20 epochs and performed early stopping based on the performance of the model on validation seen dataset.

VI. Experiments & Results

Flat Baselines. We introduce a suite of flat[1] baselines that are similar to the ones used in VLN-CE [13]: (1) **Sequence-to-Sequence (Seq2Seq)**: an encoder-decoder architecture trained using teacher-forcing [9], (2) **Progress Monitor**

[1]Flat as in there is no explicit hierarchical design for agent's decision making of high- or low-level actions.

Fig. 4: **Comparison with strong flat baselines**: Our proposed hierarchical method in comparison with strong flat baselines evaluated on the validation unseen dataset. Our approach shows superior performance and better generalization in unseen settings.

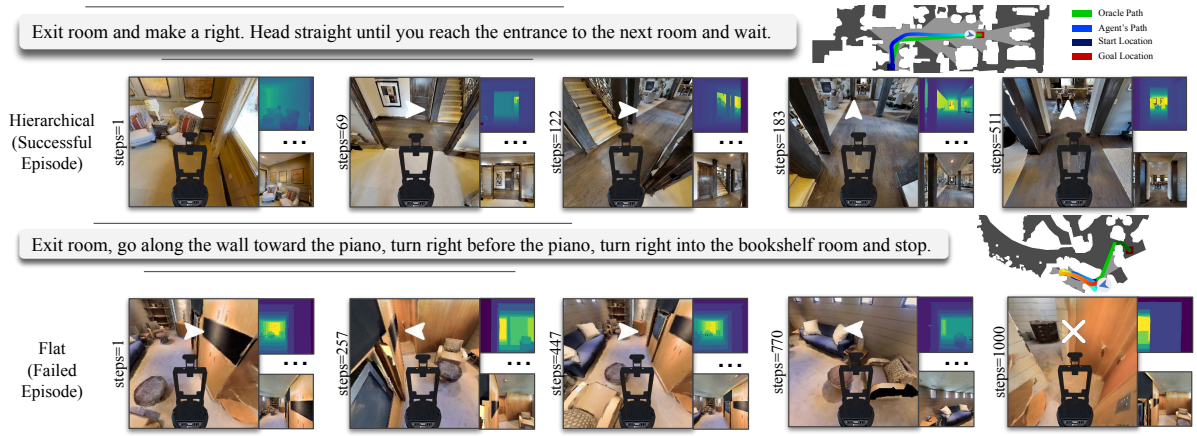

Fig. 5: **Qualitative Comparison:** Inference performance of hierarchical and flat model in unseen environments within Robo-VLN. The hierarchical model successfully predicts low-level velocity commands to reach a goal location whereas flat model bumps into obstacles.

(PM): an agent based on the Seq2Seq model but with an auxiliary loss for progress monitoring, conceptually similar to [16], and (3) **Cross-Modal Attention (CMA)**: an cross-modality attention based agent that is conceptually similar to RCM [17]. We adapt these baselines into our Robo-VLN task but with a single change: the output layers now predicts linear and angular velocities as well as the stop action, as opposed to the four actions (forward, turn-left, turn-right, and stop) used in VLN-CE. Note that baselines are without DAgger [47] and data augmentation from [48].

Comparison with Flat Baselines. The results of our proposed HCM against baselines are summarized in Table II. As shown in Table II and Figure 4, our proposed approach, which uses a hierarchical structure to tackle the long-horizon Robo-VLN problem, consistently outperforms the strong baseline models. Specifically, our HCM agent shows superior validation unseen performance by achieving a 40% SPL and 46% SR; hence demonstrating an absolute 13% improvement in SR and 10% improvement in SPL over the best performing baseline on the validation unseen environments.

Ablation Study. In our ablation experiments, we empirically validate the significance of different design choices and modules in our proposed HCM agent. Our results are summarized in Table III. First, we ablate *vision* (RGB and Depth) in our model. Our results show that an agent without vision performs as good as a random agent (*i.e.,* 0.07 SPL, 0.07SR). It shows the effectiveness of vision for end-to-end trainable agents in photo-realistic simulations. Second, we consider an architecture with early RGB and Depth fusion before cross attention with language. Our results show that separately aligning RGB and Depth with instructions performs much better than attending to the instructions corresponding to a fused RGB-D representation. We further ablate *hierarchy* to show the importance of hierarchy in our architecture. Our results are summarized as follows.

Is the source of improvement from *hierarchy*? Our method relies on decomposing the complex task into layered decision making; the top level predicts a sub-goal whereas the bottom level predicts low-level velocity commands. To confirm that hierarchy is indeed the source of improvement, we devise an experiment, in which we *flattened* the hierarchical model and provide auxiliary sub-goal supervision to the flattened model in addition to the low-level supervisions. This model effectively reduced to Seq2Seq baseline model but with high-level action supervision. The results are reported in Table III (#2 vs #4). We show that, despite using same levels of supervisions, the flattened hierarchical model under-performs the hierarchical approach, *e.g.,* 40% vs 46% in SR and 34% vs 40% in SPL. This comparison demonstrates that decoupling reasoning and imitation indeed plays a pivotal role in learning effective individual policies.

Qualitative Comparison. We qualitatively analyze the performance of hierarchical and flat agents in Robo-VLN. As shown in Figure 5, the hierarchical agent (top example) successfully predicts low-level velocity commands while reaching a desired goal location described by the instruction. The agent takes significantly more steps than discrete VLN settings (511 steps) to reach the goal location; hence showing the effectiveness of hierarchical agents to solve long horizon cross-modal trajectory following problem. The flat agent (bottom figure) fails to follow the trajectory and drives into obstacles multiple times. The episode ends after the agent is unsuccessful in reaching the goal at 1000 steps.

VII. Conclusion

Despite the recent progress, existing VLN environments impose certain unrealistic assumptions such as perfect localization, known topology and deterministic navigation in the absence of any obstacles. In this work, we first propose the Robo-VLN setting that lifts off the unrealistic assumption of navigation graph and discrete action space and provides a suite of strong baselines inspired by the recent works in discrete VLN setting. We then take the next step to propose a Hierarchical Cross-Modal (HCM) agent that tackles the challenging long-horizon issue in Robo-VLN via a hierarchical model design. Our proposed HCM agent, with trained high- and low-level policies, achieves significant performance improvement against the strong baselines. We believe that our new Robo-VLN setting and strong benchmarks would help build a stronger suite of autonomous agents.

References

[1] Arun Balajee Vasudevan, Dengxin Dai, and Luc Van Gool. Talk2nav: Long-range vision-and-language navigation with dual attention and spatial memory. *International Journal of Computer Vision*, pages 1–21, 2020.

[2] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. *arXiv preprint arXiv:2004.14973*, 2020.

[3] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13137–13146, 2020.

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[5] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[6] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Xin Wang, Qiuyuan Huang, A. Çelikyilmaz, Jianfeng Gao, Dinghan Shen, Y. Wang, William Yang Wang, and L. Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6622–6631, 2019.

[8] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 3318–3329, 2018.

[9] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Pararth Shah, Marek Fiser, Aleksandra Faust, Chase Kew, and Dilek Hakkani-Tur. Followmet: Robot navigation by following natural language directions with deep reinforcement learning. In *Third Machine Learning in Planning and Control of Robot Motion Workshop at ICRA*, 2018.

[11] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. 11 2019.

[12] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. 2020.

[14] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[15] Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. Mapping instructions to actions in 3D environments with visual goal prediction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2667–2678, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[16] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[17] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019.

[18] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[19] Michal Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaskowski. Vizdoom: A doom-based AI research platform for visual reinforcement learning. *CoRR*, abs/1605.02097, 2016.

[20] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*,

abs/1712.05474, 2017.

[21] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018.

[22] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[23] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, volume 11220 of *Lecture Notes in Computer Science*, pages 38–55. Springer, 2018.

[24] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Neural Modular Control for Embodied Question Answering. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2018.

[25] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3314–3325. Curran Associates, Inc., 2018.

[26] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[27] Xin Wang, Qiuyuan Huang, Asli Çelikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6629–6638. Computer Vision Foundation / IEEE, 2019.

[28] Haoshuo Huang, Vihan Jain, Harsh Mehta, Jason Baldridge, and Eugene Ie. Multi-modal discriminative model for vision-and-language navigation. *CoRR*, abs/1905.13358, 2019.

[29] Federico Landi, Lorenzo Baraldi, Marcella Cornia, Massimiliano Corsini, and Rita Cucchiara. Perceive, transform, and act: Multi-modal attention networks for vision-and-language navigation, 2020.

[30] R.S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[31] Richard Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[32] A. S. Vezhnevets, Simon Osindero, T. Schaul, N. Heess, Max Jaderberg, D. Silver, and K. Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. *ArXiv*, abs/1703.01161, 2017.

[33] Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudik, Yisong Yue, and Hal Daumé, III. Hierarchical imitation and reinforcement learning. volume 80 of *Proceedings of Machine Learning Research*, pages 2917–2926, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[34] Ronan Fruit and Alessandro Lazaric. Exploration-Exploitation in MDPs with Options. volume 54 of *Proceedings of Machine Learning Research*, pages 576–584, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

[35] Junha Roh, Chris Paxton, Andrzej Pronobis, Ali Farhadi, and Dieter Fox. Conditional driving from natural language instruction. In *Proceedings of the Conference on Robot Learning*, 2019.

[36] Adithyavairavan Murali, Tao Chen, Kalyan Vasudev Alwala, Dhiraj Gandhi, Lerrel Pinto, Saurabh Gupta, and Abhinav Gupta. Pyrobot: An open-source robotics framework for research and benchmarking. *arXiv preprint arXiv:1906.08236*, 2019.

[37] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems (7th Edition)*. Pearson, 2014.

[38] Gregory Dudek and Michael Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, USA, 2nd edition, 2010.

[39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[41] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[42] Peter Anderson, Angel X. Chang, Devendra Singh

Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Roshan Zamir. On evaluation of embodied navigation agents. *CoRR*, abs/1807.06757, 2018.

[43] Gabriel Ilharco Magalhaes, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeurIPS Visually Grounded Interaction and Language (ViGIL) Workshop*. 2019.

[44] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[45] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect point-goal navigators from 2.5 billion frames. *International Conference on Learning Representations (ICLR)*, 2020.

[46] Ilya Sutskever. *Training recurrent neural networks*. University of Toronto Toronto, Canada, 2013.

[47] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[48] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *arXiv preprint arXiv:1904.04195*, 2019.