LANA: A Language-Capable Navigator for Instruction Following and Generation

Xiaohan Wang Wenguan Wang Jiayi Shao Yi Yang*
CCAI, Zhejiang University

https://github.com/wxh1996/LANA-VLN

Abstract

*Recently, visual-language navigation (VLN) – entailing robot agents to follow navigation instructions – has shown great advance. However, existing literature put most emphasis on interpreting instructions into actions, only delivering "**dumb**" wayfinding agents. In this article, we devise* **LANA**, *a **language-capable** navigation agent which is able to not only execute human-written navigation commands, but also provide route descriptions to humans. This is achieved by simultaneously learning instruction following and generation with only one **single** model. More specifically, two encoders, respectively for route and language encoding, are built and shared by two decoders, respectively for action prediction and instruction generation, so as to exploit cross-task knowledge and capture task-specific characteristics. Throughout pretraining and fine-tuning, both instruction following and generation are set as optimization objectives. We empirically verify that, compared with recent advanced task-specific solutions,* LANA *attains better performances on both instruction following and route description, with nearly half complexity. In addition, endowed with language generation capability,* LANA *can explain to human its behaviours and assist human's wayfinding. This work is expected to foster future efforts towards building more trustworthy and socially-intelligent navigation robots.*

1. Introduction

Developing agents that can interact with humans in natural language while perceiving and taking actions in their environments is one of the fundamental goals in artificial intelligence. As a small step towards this target, visual-language navigation (VLN) [4] – endowing agents to execute natural language navigation commands – recently received significant attention. In VLN space, much work has been done on *language grounding* – teaching agents how to relate human instructions with actions associated with perceptions. However, there has been far little work [27, 70, 1, 77, 23] on the reverse side – *language generation* – teaching agents how to verbalize a vivid description of navigation routes. More critically, existing VLN literature separately train agents that are specialized for each single task. As a result, the delivered agents are either strong wayfinding actors but never talking, or conversable route instructors but never walking.

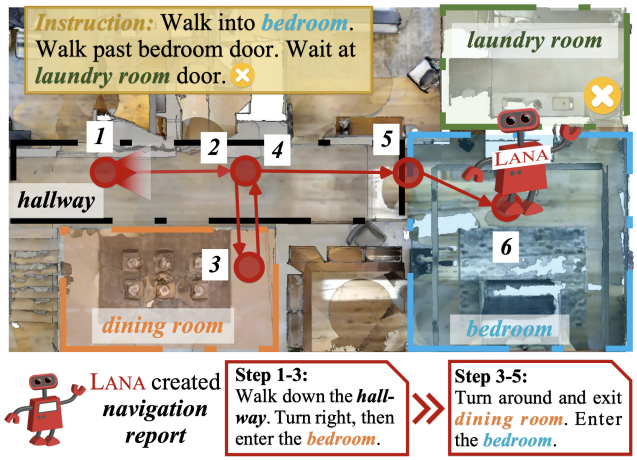

Figure 1: LANA is capable of both instruction following and generation. Its written report benefits human-robot collaboration, and, to some extend, can explain its behavior: it takes a wrong action at step 2 as it mistakes the dining room for bedroom. After gathering more information at step 3, it changes to the correct direction.

This article underlines a fundamental challenge in VLN: *Can we learn a single agent that is capable of both navigation instruction following and route description creation?*

We propose LANA, a language-capable navigation agent, that is fully aware of such challenge (Fig. 1). By simultaneously learning instruction grounding and generation, LANA formalises *human-to-robot* and *robot-to-human* communication, conveyed using navigation-oriented natural language, in a unified framework. This is of great importance, because: **i)** It completes the necessary communication cycle between human and agents, and promotes VLN agent's real-world utility [58]. For instance, when an agent takes long time to execute a navigation command, during which sustained human attention is infeasible and undesirable, the agent should report its progress [72]. Also, agents are expected to direct human in agents' explored areas [81], which is relevant for search and rescue robots in disaster regions [71, 19], guide

*Corresponding author: Yi Yang.

robots in public spaces [77], and navigation devices for the visually impaired [36]. **ii)** Two-way communication is integral to tight human-robot coordination (*i.e.*, "*I will continue this way* $\cdots$") [7], and boosts human trust in robot [6, 24], hence increasing the acceptance of navigation robots. **iii)** Developing the language generation skill makes for more explainable robots, which can interpret their navigation behaviors in a form of human-readable route descriptions.

Technically, LANA is built as a Transformer-based, multi-task learning framework. The network consists of two uni-modal *encoders* respectively for language and route encoding, and two multimodal *decoders* respectively for route-to-instruction and instruction-to-route translation, based on the two encoders. The whole network is end-to-end learned with the tasks of both instruction grounding and generation, during both pretraining and fine-tuning phases. Taken all these together, LANA provides a unified, powerful framework that explores both task-specific and cross-task knowledge at the heart of model design and network training. LANA thus can better comprehend linguistic cues (*e.g.*, words, phrases, and sentences), visual perceptions, actions over long temporal horizons and their relationships, even in the absence of explicit supervision, and eventually benefits both the two tasks.

We conduct extensive experiments on three famous VLN datasets (*i.e.*, R2R [4], R4R [38], REVERIE [62]), for both instruction following and generation, giving a few intriguing points: **First**, LANA successfully solves the two tasks using only one single agent, without switching between different models. **Second**, with an elegant and integrated architecture, LANA performs comparable, or even better than recent top-leading, task-specific alternatives. **Third**, compared to learning each task individually, training LANA on the two tasks jointly obtains better performance with reduced complexity and model size, confirming the advantage of LANA in cross-task relatedness modeling and parameter efficiency. **Forth**, LANA can explain to human its behavior by verbally describing its navigation routes. LANA can be essentially viewed as an explainable VLN robot, equipped with a self-adaptively trained language explainer. **Fifth**, subjective analyses reveal our linguistic outputs are of higher quality than the baselines but still lag behind human-generated utterances. While there is still room for improvement, our results shed light on a promising direction of future VLN research, with great potential for explainable navigation agents and robot applications.

2. Related Work

Navigation Instruction Following. Building autonomous, language-based navigation agents is a long-standing target for natural language processing and robotics communities. Rather than previous studies bounded to controlled environmental context [55, 71, 10, 5, 57], Anderson *et al*. [4] lift such task to a photo-realistic setting – VLN, stimulating increasing interest in computer vision field. Early efforts were built upon recurrent neural networks. They explore diverse training strategies [83, 82], mine extra supervisory signals from synthesized samples [27, 70, 28] or auxiliary tasks [82, 35, 53, 94, 77], and explore intelligent path planning [39, 54, 80]. For structured and long-range context modeling, recent solutions were developed with environment map [93, 13, 21, 79], transformer architectures [33, 60, 48, 63, 11], and multi-modal pretraining [56, 31, 30, 12].

Unlike existing VLN solutions that are *all* specialized for navigation instruction following, we are ambitious to build a powerful agent that is able to not only execute navigation instructions but also describe its navigation routes. We stick to this target throughout our algorithm – from network design, to model pretraining, to fine-tuning. Through jointly learning instruction execution and generation, our agent can better ground instructions into perception and action, and, to certain degree, interpret its behavior and foster human trust. Our target and visual-dialog navigation [72] are different (yet complementary), as the latter only focuses on the situation where agents use language to ask for human assistance.

Navigation Instruction Generation. The study of instruction creation [17] can date back to the 1960s [52]. Early work [87, 2, 51] found human route direction is tied to cognitive map [42], and impacted by many factors, *e.g.*, cultural background [73] and gender [37]. They also reached a consensus that involving *turn-by-turn directions* and *salient landmarks* makes instructions easier for human to follow [50, 76, 66]. Based on these efforts, a few computational systems are developed using pre-built *templates* [50, 29], or hand-crafted *rules* [18]. Though providing high-quality output in targeted scenarios, they require expertise of linguistic knowledge and extensive effort for building the templates/rules. Some data-driven solutions [16, 58, 19, 26] emerged later, yet confined to simplified grid-like or perception-poor environments.

Generating natural language instructions has long been viewed as a core functionality of socially intelligent robots and been of great interest in many disciplines such as robotics [29], linguistics [68], cognition [42, 25], psychology [73], and geo science [20]. Surprisingly little has been done in the field of embodied vision. For the rare exceptions [27, 70, 67, 1, 77, 23], [27, 70, 67] are only to augment the training data for boosting wayfinding, and, all of them learn a single agent specialized for instruction generation. Our idea is fundamentally different. We are to build a language-capable navigation agent that masters both instruction following and creation. As a result, this work represents an early yet solid attempt towards socially intelligent, embodied navigation robots.

Auxiliary Learning in VLN. There are several VLN solutions [53, 94, 78] exploit extra supervision signals from auxiliary tasks to aid navigation policy learning. For the auxiliary tasks, representative ones include next-step orientation regression [94], navigation progress estimation [53], path back-translation [94, 77], trajectory-instruction compatibility pre-

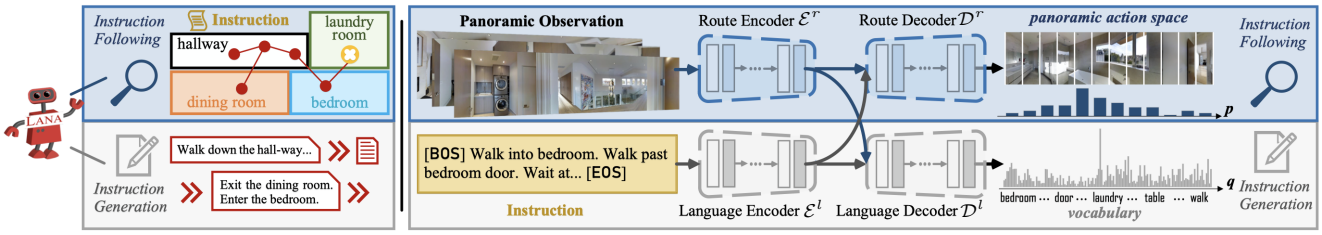

Figure 2: Architecture overview of LANA (§3.1).

diction [94], as well as final target localization [93].

These VLN solutions put focus on instruction following; the auxiliary tasks are the means, not the end. By contrast, we aim to build one single agent that learns to master both instruction following and creation well. Although [77] pays equal attention to instruction following and generation under a dual-task learning scheme, it still learns two separate single-task agents. Moreover, the aforementioned auxiliary tasks, in principle, can be utilized by our agent.

Vision-Language Pretraining for VLN. Vision-language pretraining [64, 69, 14] on massive-scale image-text pairs has recently witnessed rapid progress. It has been proven that transferable cross-modal representations can be delivered via such pretraining and facilitate downstream tasks [84, 47, 92, 64, 44, 46, 90, 89]. Such training regime has become increasingly popular in VLN. In particular, a few early endeavors [45, 33] directly adopt general vision-language pretraining for VLN, without considering task-specific nature. Later, [30, 31, 12] conduct pretraining on abundant web image-captions [30] or synthesized trajectory-instruction pairs [31, 12] with different VLN-specific proxy tasks. [11, 63] introduce history-aware proxy tasks for more VLN-aligned pretraining.

From the view of proxy task, existing VLN pretraining follows the *masked language modeling* regime [40]. Differently, our pretraining is based on language generation, which helps the agent to capture the linguistic structures so as to reach comprehensive understanding of language commands and boost instruction execution. Recent advance [86, 91, 22, 85] in general vision-language pretraining also confirm the value of generative language modeling. Moreover, for LANA, instruction generation is not merely a proxy task that is often dropped after pretraining, but also a main training target during fine-tuning, and the fundamental basis for the ability of language-based route direction during deployment.

3. Methodology

Task Setup. Our target is to build a language-capable navigation agent, which masters both instruction following and generation tasks, using one single model instance only.

- *Instruction following:* the agent needs to find a route $R=\{r_t\}_{t=1}^{T}$ of T steps to a target location, following a human-written instruction $X=\{x_l\}_{l=1}^{L}$ of L words. At each step t, the agent gets a panoramic RGB percept O_t, discretized into $K=36$ views, *i.e.*, $O_t=\{o_{t,k}\in\mathbb{R}^{3\times224\times224}\}_{k=1}^{K}$. The agent selects a navigation action a_t, *i.e.*, a navigable view, from $\{o_{t,k}\}_k$ to perform. Together, O_t and a_t determine the route state r_t at step t, *i.e.*, $r_t=(O_t, a_t)$.
- *Instruction generation:* the agent observes a navigation route $R=\{r_t\}_{t=1}^{T}$, *i.e.*, a sequence of actions $\{a_t\}_{t=1}^{T}$ along with the panoramic percept $\{O_t\}_{t=1}^{T}$, and must verbalize a grounded description $X=\{x_l\}_{l=1}^{L}$ of the route R.

Method Overview. To address our challenging setting, we devise LANA, a Transformer-based, multi-task VLN agent (Fig. 2) that exploits cross-task commonalities throughout architecture design (§3.1) and network training (§3.2).

3.1. Model Architecture

LANA can take the navigation route $R=\{r_t\}_t$ as input and output corresponding description $X=\{x_l\}_l$, and vice versa. To achieve such *bi-directional translation* between navigation route R and instruction X, and explore cross-task relatedness, LANA is elaborately designed as a composition of:

- *Route Encoder* $\mathcal{E}^r$ and *Language Encoder* $\mathcal{E}^l$ for unimodality representation encoding based on self-attention; and
- *Language Decoder* $\mathcal{D}^l$ and *Route Decoder* $\mathcal{D}^r$ for cross-attention based route-language bi-directional translation.

The two unimodal encoders are shared between and jointly trained with the two multimodal decoders. They work closely:

- *Instruction Following:* at navigation step t, LANA respectively feeds the entire instruction X and the sequence of historical route states $\{r_1,\cdots,r_{t-1}\}$ and current percept O_t into the corresponding encoders, and utilizes the route decoder $\mathcal{D}^r$ to predict the navigation action a_t.
- *Instruction Generation:* at generation step l, LANA respectively feeds the full route R and prior predicted words $\{x_1,\cdots,x_{l-1}\}$ into the corresponding encoders, and utilizes the language decoder $\mathcal{D}^l$ to predict the next word x_l.

We remark that LANA conducts the instruction generation task in an *autoregressive* manner. In this way, both the two decoders are conditioned on both the two encoders, leading to extensive visual and linguistic knowledge exchange.

Route Encoder $\mathcal{E}^r$. $\mathcal{E}^r$ takes input either the entire route, *i.e.*, $R=\{r_t\}_{t=1}^{T}=\{(O_t,a_t)\}_{t=1}^{T}$, during instruction generation; or historical route states along with current observation, *i.e.*, $\{r_1,\cdots,r_{t-1},O_t\}=\{O_1,a_1,\cdots,O_{t-1},a_{t-1},O_t\}$, during wayfinding. Hence it has two types of input tokens corresponding to the panoramic observation O_t and action a_t. In particular, the observation token of O_t is calculated by:

$$\begin{aligned}\boldsymbol{O}_t &= [\boldsymbol{o}_{t,1},\boldsymbol{o}_{t,2},\cdots,\boldsymbol{o}_{t,K}] \quad \in\mathbb{R}^{K\times d},\\ \boldsymbol{o}_{t,k} &= \mathcal{F}^v(\boldsymbol{v}_{t,k})+\mathcal{F}^\theta(\boldsymbol{\theta}_{t,k})+\boldsymbol{\tau}^t+\boldsymbol{\tau}^O \quad \in\mathbb{R}^d,\end{aligned} \tag{1}$$

where $\boldsymbol{v}_{t,k}$ and $\boldsymbol{\theta}_{t,k}$ are respectively the visual and orientation embeddings of view $o_{t,k}$; $\mathcal{F}^{v/\theta}$ is linear projection for feature dimension alignment; $\boldsymbol{\tau}^t \in \mathbb{R}^d$ embeds the temporal order – t; and $\boldsymbol{\tau}^O \in \mathbb{R}^d$ is a learnable type embedding which indicates $\boldsymbol{o}_{t,k}$ is an observation token.

Similarly, the action token of a_t is given as:

$$\boldsymbol{a}_t = \mathcal{F}^v(\boldsymbol{v}_{t,a_t}) + \mathcal{F}^\theta(\boldsymbol{\theta}_{t,a_t}) + \boldsymbol{\tau}^t + \boldsymbol{\tau}^A \ \in \mathbb{R}^d, \quad (2)$$

where $\boldsymbol{v}_{t,a_t}$ and $\boldsymbol{\theta}_{t,a_t}$ respectively embed the visual view and turned angle that are associated with action a_t. Analogous to Eq. 1, $\boldsymbol{\tau}^A \in \mathbb{R}^d$ encodes the action token type.

Tokenizing all the K subviews $\{\boldsymbol{o}_{t,k}\}_k$ of each panoramic percept O_t allows LANA to access/memorize all the observations along the navigation route. Unfortunately, considering such many tokens causes unaffordable computation load for self-attention based encoding. To pursue a good balance between computational cost and representation ability, we first compute an action-attentive route state:

$$\begin{aligned} &\boldsymbol{r}_t = \boldsymbol{a}_t + \boldsymbol{c}_t \ \in \mathbb{R}^d, \\ &\boldsymbol{c}_t = cross\_att(\boldsymbol{a}_t, \boldsymbol{O}_t) = cross\_att(\boldsymbol{a}_t, [\boldsymbol{o}_{t,k}]_{k=1}^K) \ \in \mathbb{R}^d. \end{aligned} \quad (3)$$

Through cross-attention, *i.e.*, $cross\_att(\cdot,\cdot)$, action-related visual context $\boldsymbol{c}_t$ are gathered and compressed into a d-dimensional vector. Then the output of $\mathcal{E}^r$ is obtained via:

$$\begin{aligned} &\text{Ins. following: } [\bar{\boldsymbol{r}}_{1:t-1}, \bar{\boldsymbol{O}}_t] = self\_att([\boldsymbol{r}_{1:t-1}, \boldsymbol{O}_t]) \in \mathbb{R}^{(t-1+K)\times d}, \\ &\text{Ins. generation: } \quad [\bar{\boldsymbol{r}}_{1:T}] = self\_att([\boldsymbol{r}_{1:T}]) \in \mathbb{R}^{T\times d}. \end{aligned} \quad (4)$$

Language Encoder $\mathcal{E}^l$. $\mathcal{E}^l$ takes input either the complete instruction, *i.e.*, $X = \{x_l\}_{l=1}^L$, during wayfinding; or previously generated words, *i.e.*, $\{x_1, \cdots, x_{l-1}\}$, during instruction generation. It is built as a standard Transformer language encoder for contextualized linguistic feature extraction:

$$\begin{aligned} &\text{Ins. following: } \quad [\bar{\boldsymbol{x}}_{1:L}] = \mathcal{E}^l([x_{1:L}]) \in \mathbb{R}^{L\times d}, \\ &\text{Ins. generation: } [\bar{\boldsymbol{x}}_{1:l-1}] = \mathcal{E}^l([x_{1:l-1}]) \in \mathbb{R}^{(l-1)\times d}, \end{aligned} \quad (5)$$

where $\mathcal{E}^l$ contains several blocks, each of which has a multi-head self-attention layer and a feed-forward sub-layer [65]; position embeddings are omitted for the sake of brevity.

We note that, during the training of the instruction generation task, *causal future mask* [74] is applied to each self-attention layer, ensuring each word token can only attend to the previous ones, and allowing our single model to tackle both instruction following and generation simultaneously.

Route Decoder $\mathcal{D}^r$. $\mathcal{D}^r$ is for instruction-to-route translation. Concretely, at navigation step t, $\mathcal{D}^r$ takes input the complete instruction embedding $\bar{\boldsymbol{x}}_{1:L}$, historical route states $\bar{\boldsymbol{r}}_{1:t-1}$, as well as current observation feature $\bar{\boldsymbol{O}}_t = \bar{\boldsymbol{o}}_{t,1:K}$, and outputs probability distribution $\boldsymbol{p}_t \in \triangle^K$ of action selection over current K subviews $o_{t,1:K}$[1]. More specifically, $\mathcal{D}^r$ is built as a stack of several cross-attention-based blocks for modeling cross-modal relationships. For each block, we have:

$$[\hat{\boldsymbol{r}}_{1:t-1}, \hat{\boldsymbol{o}}_{t,1:K}] = cross\_att([\bar{\boldsymbol{r}}_{1:t-1}, \bar{\boldsymbol{o}}_{t,1:K}], \bar{\boldsymbol{x}}_{1:L}), \quad (6)$$

$$[\bar{\boldsymbol{r}}_{1:t-1}, \bar{\boldsymbol{o}}_{t,1:K}] \leftarrow self\_att([\hat{\boldsymbol{r}}_{1:t-1}, \hat{\boldsymbol{o}}_{t,1:K}]). \quad (7)$$

Eq. 6 obtains language-enhanced route and observation representations, *i.e.*, $\hat{\boldsymbol{r}}_{1:t-1}$ and $\hat{\boldsymbol{o}}_{t,1:K}$, through cross-attention. Eq. 7 adopts self-attention to model temporal dependencies among historical route states $\hat{\boldsymbol{r}}_{1:t-1}$, and capture the correlations between $\hat{\boldsymbol{r}}_{1:t-1}$ and current observation $\hat{O}_t = \hat{\boldsymbol{o}}_{t,1:K}$.

After several $\mathcal{D}^r$ decoder blocks, the action probability over the K subviews $o_{t,1:K}$ is given as:

$$\boldsymbol{p}_t = softmax(\{\mathcal{F}^r(\bar{\boldsymbol{o}}_k)\}_{k=1}^K) \ \in \triangle^K, \quad (8)$$

where $\mathcal{F}^r : \mathbb{R}^d \to \mathbb{R}$ is a two-layer feed-forward network for action score mapping, as in [11, 93].

Language Decoder $\mathcal{D}^l$. $\mathcal{D}^l$ is for route-to-instruction translation. Concretely, at instruction generation step l, $\mathcal{D}^l$ takes input the full route states $\bar{\boldsymbol{r}}_{1:T}$, and the embeddings of previously generated instruction words $\bar{\boldsymbol{x}}_{1:l-1}$, and outputs probability distribution $\boldsymbol{q}_l \in \triangle^M$ of word selection over a predefined vocabulary with M words. Analogous to $\mathcal{D}^r$, $\mathcal{D}^l$ has several cross-attention based blocks. Each block is given as:

$$\hat{\boldsymbol{x}}_{1:l-1} = cross\_att(\bar{\boldsymbol{x}}_{1:l-1}, \bar{\boldsymbol{r}}_{1:T}), \quad (9)$$

$$\bar{\boldsymbol{x}}_{1:l-1} \leftarrow causal\_self\_att(\hat{\boldsymbol{x}}_{1:l-1}). \quad (10)$$

Eq. 9 lets the text attend to the route context. In Eq. 10, we adopt the *causally-masked self-attention*, instead of normal, bi-directional self-attention, to force $\mathcal{D}^l$ to "attend-ahead", which is needed for autoregressive inference.

After several $\mathcal{D}^l$ decoder blocks, the probability over the M-word vocabulary is given as:

$$\boldsymbol{q}_l = softmax(\mathcal{F}^l(\bar{\boldsymbol{x}}_{l-1})) \ \in \triangle^M, \quad (11)$$

where $\mathcal{F}^r : \mathbb{R}^d \to \mathbb{R}^M$ is a two-layer feed-forward network for the prediction of the word score distribution.

3.2. Network Training

All the modules of LANA, *i.e.*, two unimodal encoders $\mathcal{E}^r$ and $\mathcal{E}^l$, as well as two multimodal decoders $\mathcal{D}^r$ and $\mathcal{D}^l$, are jointly end-to-end learned, by optimizing the training objectives of instruction following and generation.

Instruction Generation. For each instruction-route training pair (X, R), where $X = x_{1:L}$ and $R = r_{1:T}$, LANA learns instruction generation by predicting x_l based on the full route R and proceeding reference words $x_{0:l-1}$. We append two special tokens to X, *i.e.*, $x_0 = [\texttt{BOS}]$ and $x_{L+1} = [\texttt{EOS}]$, respectively indicating the start and end of the instruction sentence. To generate word x_l, LANA respectively feeds R and $x_{0:l-1}$ into $\mathcal{E}^r$ and $\mathcal{E}^l$ for unimodal encoding (*cf.* Eq. 4&5). Conditioned on the route and linguistic embeddings, *i.e.*, $\bar{\boldsymbol{r}}_{1:T}$ and $\bar{\boldsymbol{x}}_{1:l-1}$, $\mathcal{D}^l$ gives the word probability $\boldsymbol{q}_l$ (*cf.* Eq. 11). The training objective of instruction generation, formulated as the language modeling loss, can be written as:

$$\mathcal{L}^g = -\sum_{l=1}^{L+1} \log(p(x_l | x_{0:l-1}, R)) = -\sum_{l=1}^{L+1} \log(\boldsymbol{q}_l(x_l)), \quad (12)$$

[1] Note that, in addition to the K action subviews, `STOP` token is also considered here, leading to $K+1$ decision choices. We omit `STOP` for simplicity.

where $\boldsymbol{q}_l(x_l) \in [0,1]$ is the probability of word x_l. LANA is trained to minimize the negative log-likelihood of the reference instruction words. Teacher-forcing [88] is used here to enable the parallel text input. Worth mentioning is that, existing VLN pretraining methods [30, 31, 30, 11, 63] rely on the masked language modeling (MLM) strategy. Since MLM only predicts a small portion (typically 15%) of input words during each training iteration, it is less efficient for large-scale pretraining data, as pointed out by many recent literature in general vision-language pretraining [34, 9, 15].

Instruction Following. For each training pair (X, R), where $X = x_{1:L}$ and $R = r_{1:T} = (O_t, a_t)_{1:T}$, LANA concurrently learns instruction following by predicting a_t based on the full instruction X, history from expert demonstration $r_{1:t-1}$, and the current percept O_t. Specifically, LANA respectively feeds X and $\{r_{1:t-1}, O_t\}$ into $\mathcal{E}^l$ and $\mathcal{E}^r$ (*cf.* Eq. 4&5). Conditioned on the output unimodal encodings, *i.e.*, $\bar{\boldsymbol{x}}_{1:L}$ and $[\bar{\boldsymbol{r}}_{1:t-1}, \bar{\boldsymbol{O}}_t]$, $\mathcal{D}^r$ gives the action probability $\boldsymbol{p}_t$ (*cf.* Eq. 8). The training objective of instruction following is to minimize the negative log-likelihood of the target view action a_t:

$$\mathcal{L}^f = -\sum_{t=1}^{T} \log(p(a_t | r_{0:t-1}, O_t, X)) = -\sum_{t=1}^{T} \log(\boldsymbol{p}_t(a_t)). \quad (13)$$

LANA is end-to-end learned with the two training targets (*cf.* Eq. 12&13) during both pretraing and fine-tuning phases. Note that the encoders $\mathcal{E}^r$ and $\mathcal{E}^l$ receive the supervision signals from both instruction generation (*cf.* Eq. 12) and following (*cf.* Eq. 13). Moreover, such a joint learning framework grants LANA improved interpretability – LANA can be viewed as a navigator born with a language explainer $\mathcal{D}^l$.

3.3. Implementation Details

Network Architecture. The route $\mathcal{E}^r$ and language $\mathcal{E}^l$ encoders respectively have one and nine layers, and the decoders $\mathcal{D}^r$ and $\mathcal{D}^l$ both have four blocks. The feature dimension is set as $d = 768$. The orientation feature $\boldsymbol{\theta}_k$ of view o_k (*cf.* Eq. 1) is defined as: $\boldsymbol{\theta}_k = (\cos\varphi_k, \sin\varphi_k, \cos\phi_k, \sin\phi_k)$, where φ and ϕ are the angles of heading and elevation, respectively.

Training. Following recent VLN practice [56, 31, 30, 11, 63], the pretraining and fine-tuning paradigm is adopted:

- *Pretraining*: With the two training objectives (*cf.* Eq. 12& 13), LANA is pretrained on offline-sampled instruction-route pairs from PREVALENT [31], including 104K original R2R samples and 6482K synthesized ones. LANA is trained for 100K iterations, using Adam optimizer [41] with 1e-4 learning rate, and $N = 128$ batch size.
- *Fine-tuning*: Then we fine-tune LANA on different VLN datasets, still using our two training tasks (*cf.* Eq. 12&13). Following the standard protocol [33, 77, 32, 82], the training of instruction following is based on the mixture of *imitation learning* and *reinforcement learning*. In this stage, we set the learning rate to 1e-5 and batch size to 8.

We use four NVIDIA Tesla A100 GPUs for network training, and sample only one training task for each mini-batch.

Inference. Once trained, LANA is capable of both following and verbalizing navigation instructions with only one single model instance, without any architectural change. Specifically, for instruction following, greedy search, *i.e.*, selecting the action with the highest probability at each prediction step, is adopted and terminated when `STOP` is chosen. For instruction generation, the sentence is predicted in an autoregressive manner, *i.e.*, generating one word at a time until `EOS` is chosen, conditioned on previous generated ones.

4. Experiment

We evaluate LANA for both instruction following (§4.1) and generation (§4.2) tasks, followed by a series of diagnostic experiments (§4.3) and qualitative studies (§4.4).

For each task, we give scores of two versions of LANA:

- LANA$_{mt}$: jointly learn the two target tasks throughout pretraining and fine-tuning. Thus, such multi-task version only has one single agent instance, tested on the two tasks.
- LANA$_{st}$: jointly pre-train on the two tasks, but fine-tune on each task individually. There are two single-task agent instances; each is only tested on the corresponding task.

We collect here key observations from our subsequent experiments: **i)** LANA performs comparable, or even better than prior tasks-specific agents; **ii)** LANA$_{mt}$ outperforms LANA$_{st}$ with more efficient parameter utilization; **iii)** LANA can provide test-time behavioral interpretation by verbalizing descriptions of its navigation routes; and **iv)** our model design and training targets indeed contribute to our strong results.

4.1. Performance on Instruction Following

Dataset. We conduct experiments on three VLN datasets:

- R2R [4]: It has four splits, *i.e.*, `train` (61 scenes, 14,039 instructions), `val seen` (61 scenes, 1,021 instructions), `val unseen` (11 scenes, 2,349 instructions), and `test unseen` (18 scenes, 4,173 instructions). There are no overlapping scenes between `train` and `unseen` splits.
- R4R [38]: It extends R2R by connecting two close tail-to-head trajectories and corresponding instructions in R2R. R4R contains three sets, *i.e.*, `train` (61 scenes, 233,613 instructions), `val seen` (61 scenes, 1,035 instructions), and `val unseen` (11 scenes, 45,162 instructions).
- REVERIE [62]: It replaces detailed instructions in R2R with high-level descriptions of target locations and objects. It is composed of four sets, *i.e.*, `train` (53 scenes, 10,466 instructions), `val seen` (61 scenes, 1,371 instructions), `val unseen` (10 scenes, 3,753 instructions), and `test unseen` (16 scenes, 6,292 instructions).

Evaluation Metric. For R2R, we follow conventions [4, 27] to report four evaluation metrics: i) *Success Rate* (`SR`), ii) *Trajectory Length* (`TL`), iii) *Oracle success Rate* (`OR`), and iv) *Success rate weighted by Path Length* (`SPL`), where `SR` and `SPL` are of priority. For R4R, we further adopt v) *Coverage weighted by Length Score* (`CLS`) [38], vi) *normalized*

Methods	R2R val unseen				R2R test unseen			
	SR↑	**SPL**↑	OR↑	TL↓	**SR**↑	**SPL**↑	OR↑	TL↓
BT-follower [27] [NeurIPS2018]	36	-	45	-	35	28	44	14.8
EDrop-follower [70] [NAACL2019]	52	48	-	10.7	51	47	59	11.7
AuxRN [94] [CVPR2020]	55	50	62	-	55	51	62	-
PREVALENT [31] [CVPR2020]	58	53	-	10.2	54	51	-	10.5
VLN⟳BERT [33] [CVPR2021]	63	57	-	12.0	63	57	-	12.4
AirBERT [30] [ICCV2021]	62	56	-	11.8	62	57	-	12.4
HAMT [11] [NeurIPS2021]	65	59	-	11.9	63	58	-	12.7
HOP [63] [CVPR2022]	64	57	-	12.3	64	59	-	12.7
LANA$_{st}$ **(ours)**	66	60	73	11.9	64	59	71	12.4
LANA$_{mt}$ **(ours)**	**68**	**62**	**76**	12.0	**65**	**60**	**71**	12.6

Table 1: Quantitative comparison results (§4.1) for **instruction following** on R2R [4]. '–': unavailable statistics.

Methods	R4R val unseen				
	CLS↑	nDTW↑	SDTW↑	SR↑	TL↓
BT-follower [27] [NeurIPS2018]	30	-	-	24	19.9
RCM [82] [CVPR2019]	35	30	13	26	28.5
PTA [43] [NeurIPS2021]	37	32	10	24	17.7
EDrop-follower [70] [NAACL2019]	34	-	9	29	27.0
OAAM [61] [ECCV2020]	40	-	11	31	13.8
ActiveVLN [80] [ECCV2020]	59	44	22	32	19.7
EGP [21] [NeurIPS2020]	44	37	18	30	18.3
HAMT [11] [NeurIPS2021]	57.7	50.3	31.8	44.6	-
LANA$_{st}$ **(ours)**	58.6	51.9	31.4	43.0	22.7
LANA$_{mt}$ **(ours)**	**59.7**	**52.3**	31.7	43.2	22.1

Table 2: Quantitative comparison results (§4.1) for **instruction following** on R4R [38].

Methods	REVERIE val unseen						REVERIE test unseen					
	SR↑	**SPL**↑	OR↑	TL↓	RGS↑	RGSPL↑	**SR**↑	**SPL**↑	OR↑	TL↓	RGS↑	RGSPL↑
RCM [82] [CVPR2019]	9.29	6.97	14.23	11.98	4.89	3.89	7.84	6.67	11.68	10.60	3.67	3.14
VLN⟳BERT [33] [CVPR2021]	30.67	24.90	35.02	16.78	18.77	15.27	29.61	23.99	32.91	15.86	16.50	13.51
AirBERT [30] [ICCV2021]	27.89	21.88	34.51	18.71	18.23	14.18	30.28	23.61	34.20	17.91	16.83	13.28
HAMT [11] [NeurIPS2021]	32.95	30.20	36.84	14.08	18.92	17.28	30.40	26.67	33.41	13.62	14.88	13.08
HOP [63] [CVPR2022]	30.39	25.10	35.30	17.16	18.23	15.31	29.12	23.37	32.26	17.05	17.13	13.90
LANA **(ours)**	**34.00**	29.26	**38.54**	16.28	**19.03**	16.18	**33.50**	**26.89**	**36.41**	16.75	**17.53**	**14.25**

Table 3: Quantitative comparison results (§4.1) for **instruction following** on REVERIE [62].

Methods	R2R val seen						R2R val unseen					
	SPICE↑	Bleu-1↑	Bleu-4↑	CIDEr↑	Meteor↑	Rouge↑	**SPICE**↑	Bleu-1↑	Bleu-4↑	CIDEr↑	Meteor↑	Rouge↑
BT-speaker [27] [NeurIPS2018]	0.203	0.537	0.155	0.121	0.233	0.350	0.188	0.522	0.142	0.114	0.228	0.346
EDrop-speaker [70] [NAACL2019]	0.202	-	0.245	0.493	0.228	0.467	0.181	-	0.237	0.422	0.225	0.458
VLS [1] [CVPRW2019]	0.214	0.549	0.157	0.137	0.228	0.352	0.197	0.548	0.159	0.132	0.231	0.357
CCC-speaker [77] [CVPR2022]	0.231	0.728	0.287	0.543	0.236	0.493	0.214	0.708	0.272	0.461	0.231	0.477
LANA$_{st}$ **(ours)**	0.251	0.743	0.305	0.522	0.243	0.502	0.223	0.722	0.287	0.433	0.235	0.490
LANA$_{mt}$ **(ours)**	**0.256**	**0.759**	**0.314**	0.533	**0.245**	**0.503**	**0.226**	**0.736**	**0.298**	0.457	**0.238**	**0.498**

Table 4: Quantitative comparison results (§4.2) for **instruction generation** on R2R [4].

Dynamic Time Warping (nDTW), and vii) *Success weighted by nDTW* (SDTW). For REVERIE, the first four metrics are also employed for its navigation sub-task, and viii) *Remote Grounding Success rate* (RGS) and ix) *RGS weighted by Path Length* (RGSPL) are additionally used for overall performance evaluation.

Quantitative Result. Several famous and recent advanced solutions [27, 70, 94, 82, 80, 31, 33, 21, 30, 11, 77, 63] for instruction following are involved in comparison. Note that we report the score of the single model under the **single run** setup following the tradition [33, 77, 63, 11]. As shown in Table 1, LANA$_{st}$, which is only fine-tuned on wayfinding after multi-task pretraining, demonstrates comparable, if not better, results than those alternatives on R2R. Remarkably, LANA$_{mt}$, which learns to interpret navigation paths alongside following instructions, even yields better navigation performance. For instance, LANA$_{mt}$ lifts LANA$_{st}$ by 2% and 1% SPL, on val unseen and test respectively. This verifies the efficacy of our language-capable navigation scheme and multi-task learning strategy. More significant improvements can be observed on R4R (*cf.* Table 2) and REVERIE (*cf.* Table 3), where the former focuses on long-horizon navigation with longer instructions and trajectories, while the latter gives abstract instructions only. These results confirm our generality and versatility. It is important to note that all the competitors are only aware of wayfinding, while our agent can generate grounded route descriptions for interpreting its navigation behaviors/plans.

4.2. Performance on Instruction Generation

Dataset. We compare machine generated route descriptions with the human-written instructions, on two VLN datasets:

- R2R [4]: As R2R test is preserved for benchmarking instruction following agents, we report the performance of instruction generation on val sets. Each R2R navigation path is associated with three ground-truth instructions.
- R4R [38]: Performance is reported on R4R val sets, where each path corresponds to nine ground-truth instructions.

REVERIE [62] is not involved as its instructions are high-level, concise descriptions of remote objects, which cannot serve our purpose of grounded instruction generation.

Evaluation Metric. Following [1, 77], we opt for five text metrics: i) BLEU [59], ii) CIDEr [75], iii) METEOR [8], iv) ROUGE [49], and v) SPICE [3]. For each navigation path, the metrics are averaged over all the corresponding groundtruth instructions. SPICE is considered as the primary metric.

Quantitative Result. We compare LANA with four instruction generation algorithms [27, 70, 1, 77]. Table 4 and Table 5 summarize our comparison results. We can find that our task-specific agent, *i.e.*, LANA$_{st}$, already outperforms all

Methods	R4R val seen						R4R val unseen					
	SPICE↑	Bleu-1↑	Bleu-4↑	CIDEr↑	Meteor↑	Rouge↑	**SPICE**↑	Bleu-1↑	Bleu-4↑	CIDEr↑	Meteor↑	Rouge↑
BT-speaker [27] [NeurIPS2018]	0.164	0.691	0.223	0.099	0.213	0.453	0.207	0.387	0.088	0.139	0.172	0.359
EDrop-speaker [70] [NAACL2019]	0.209	0.750	0.281	0.216	0.245	0.473	0.218	0.433	0.106	0.200	0.187	0.363
CCC-speaker [77] [CVPR2022]	0.219	0.758	0.312	0.245	0.252	0.480	0.233	0.403	0.115	0.206	0.193	0.365
LANA$_{st}$ (**ours**)	0.237	0.768	0.327	0.264	**0.265**	0.483	0.259	0.437	0.123	0.216	0.199	0.375
LANA$_{mt}$ (**ours**)	**0.245**	**0.772**	**0.333**	**0.287**	0.261	**0.484**	**0.262**	**0.443**	**0.128**	**0.231**	**0.200**	**0.376**

Table 5: Quantitative comparison results (§4.2) for **instruction generation** on R4R [38].

#	Pretraining		Fine-tuning		Instruction Following				Instruction Generaion					
	Instruction Following	*Instruction Generation*	*Instruction Following*	*Instruction Generation*	**SR**↑	**SPL**↑	OR↑	TL↓	**SPICE**↑	Bleu-1↑	Bleu-4↑	CIDEr↑	Meteor↑	Rouge↑
1			✔		52.1	48.3	59.3	11.2	-	-	-	-	-	-
2				✔	-	-	-	-	0.178	0.692	0.241	0.321	0.216	0.463
3			✔	✔	53.1	48.9	60.9	11.6	0.182	0.704	0.245	0.304	0.219	0.469
4	✔		✔		61.3	55.4	69.7	12.0	-	-	-	-	-	-
5		✔		✔	-	-	-	-	0.215	0.718	0.255	0.378	0.230	0.472
6	✔	✔	✔		65.7	59.9	73.4	11.9	-	-	-	-	-	-
7	✔	✔		✔	-	-	-	-	0.223	0.722	0.287	0.433	0.235	0.490
8	✔	✔	✔	✔	**67.9**	**61.6**	**75.7**	12.0	**0.226**	**0.736**	**0.298**	**0.457**	**0.238**	**0.498**

Table 6: **Ablation study** (§4.3) on R2R val unseen [4].

the competitors, across all the metrics and datasets. Note that, CCC [77], a current top-leading solution, learns the instruction generation model with the aid of a separate wayfinder. More impressively, our multi-task agent, *i.e.*, LANA$_{mt}$, performs on par or even better than LANA$_{st}$, demonstrating the algorithmic and functional advantages of our approach.

User Study. To provide a complete measure of the quality of our created instructions, we conduct a set of human evaluation experiments, based on pair-wise comparison. Concretely, 50 college students are asked to respectively compare the instructions generated by LANA$_{mt}$ with those created by CCC, BT-Speaker, and humans, for 100 paths in total. The paths are sampled from R2R val unseen. Finally, LANA receive more preference votes, *i.e.*, **63.4**% *vs* CCC 36.6%, and **75.1**% *vs* BT-Speaker 24.9%. Yet, human-written instructions are far more favorable, *i.e.*, 69.3% *vs* LANA 30.7%, demonstrating there remains large room for improvement.

4.3. Diagnostic Experiment

To thoroughly study the effectiveness of our language-capable navigation framework, we carry out a series of diagnostic experiments on val unseen set of R2R [4], for both instruction following and generation tasks. The experimental results are summarized in Table 6. More specifically, a total of eight baselines are involved in our ablation study:

1. fine-tune on instruction following only, *w/o* pretraining;
2. fine-tune on instruction generation only, *w/o* pre-training;
3. fine-tune on both instruction following and generation, *w/o* pretraining;
4. pretrain and fine-tune on instruction following only;
5. pretrain and fine-tune on instruction generation only;
6. pretrain on both instruction following and generation, and fine-tune on instruction following only;
7. pretrain on both instruction following and generation, and fine-tune on instruction generation only;
8. pretrain and fine-tune on both instruction following and generation.

These baselines can be roughly grouped into three classes: **i)** baselines 1,2, and 3 are all *w/o* pretraining, and fine-tune on each task either individually or jointly; **ii)** baselines 4 and 5 pretrain and fine-tune on each task individually; and **iii)** baselines 6, 7, and 8 are *w/* joint-task pretraining, and fine-tune on each task either individually or jointly. Baselines 6 and 7 are the two sing-task agents, *i.e.*, LANA$_{st}$, and baseline 8 are our finally delivered agent LANA$_{mt}$; their performance have been thoroughly reported in §4.1 and §4.2.

Also, note that baselines 1, 4, and 6 only master wayfinding, while baselines 2, 5, and 7 can only undertake the route description task. Baselines 3 and 8 are capable of both.

Several essential conclusions can be drawn:

- Joint-task fine-tuning can benefit the performance of both tasks (baseline 3 *vs* 1 *vs* 2);
- Joint-task pretraining and fine-tuning can benefit the performance of both tasks (baseline 8 *vs* 6 *vs* 7);
- Pretraining can facilitate the final performance (baseline 4 *vs* 1, baseline 5 *vs* 2, baseline 6 *vs* 1, baseline 7 *vs* 2, and baseline 8 *vs* 3);
- Joint-task pretraining is more favored than single-task pretraining (baseline 8 *vs* 4 *vs* 5);
- Joint-task pretraining and fine-tuning is more favored than all the other training strategies (baseline 8 *vs* 1-7).

Note that, joint-tasking pretraining and fine-tuning not only promotes the performance, but increases parameter efficiency, *i.e.*, baseline 8 (143 M) *vs* 6 + 7 (220 M = 123 M + 97 M). In a nutshell, our ablative experiments solidly verify the power of our idea, the efficacy of our algorithmic design, and our advantage in efficient-parameter utilization.

4.4. Qualitative Experiment

Fig. 3 depicts three exemplar navigation episodes from val unseen set of R2R [4] . Fig. 3 (a) compares LANA$_{mt}$ against LANA$_{st}$ on the instruction following task. As seen,

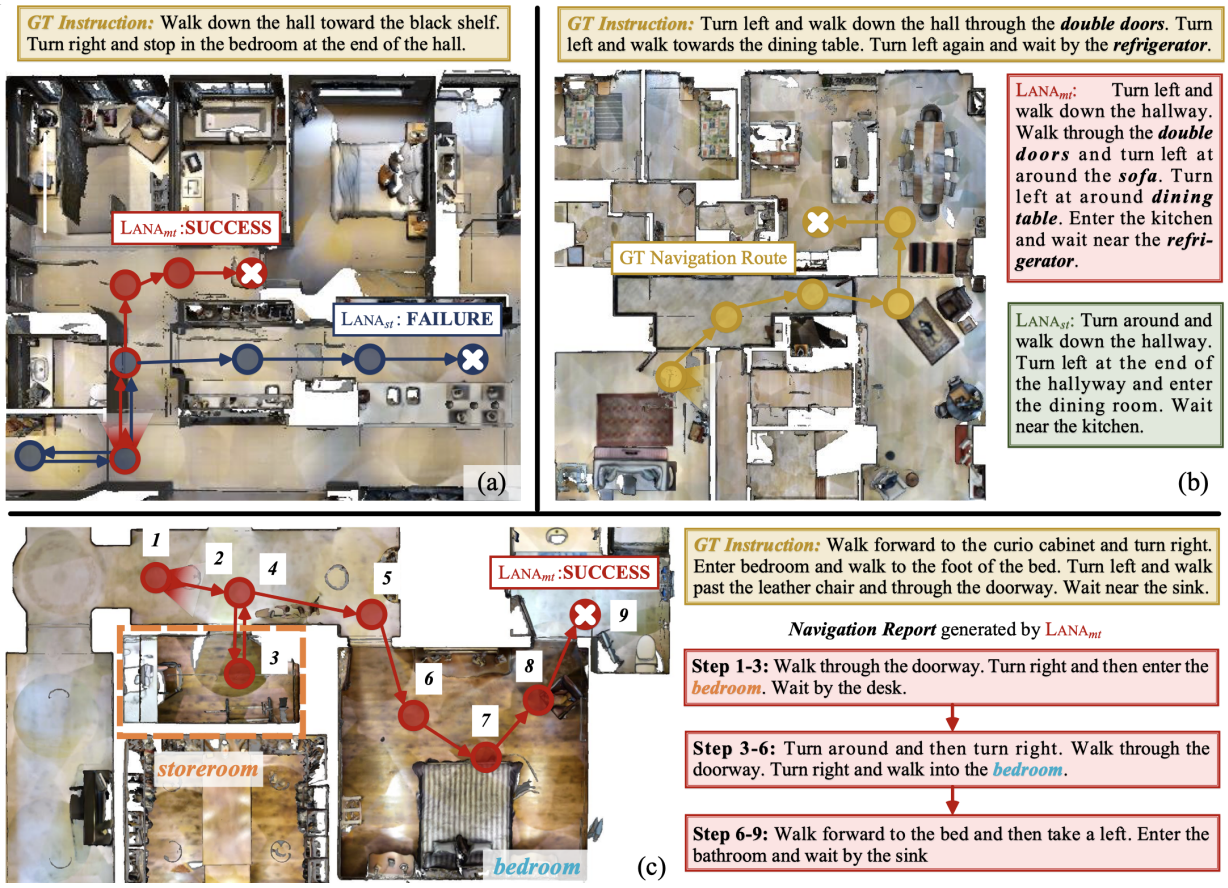

Figure 3: (a-b) Visual comparison results between LANA$_{mt}$ and LANA$_{st}$ for (a) instruction following and (b) instruction generation tasks. The start and end points of a navigation route are respectively denoted by and . (c) LANA is able to interpret its navigation behavior using natural language. The generated report keeps the monitor updated on the navigation process, and even helps reveal the failure mode. For example, at step 2-3, LANA$_{mt}$ enters the storeroom because LANA$_{mt}$ thought it is a bedroom. See §4.4 for more detailed discussion.

LANA$_{mt}$ performs robust in this challenging case, while LANA$_{st}$ fails to reach the target location. As both LANA$_{mt}$ and LANA$_{st}$ are built with similar network architectures and pretraining protocol, we attribute this to the exploration of cross-task knowledge during fine-tuning. Fig. 3 (b) visualizes comparison on instruction creation. As seen, LANA$_{mt}$ outputs more grounded instructions that contain precise action descriptions (*e.g.*, turn left, walk down) as well as salient landmarks (*e.g.*, double doors, dining table, refrigerator). These descriptions have similar properties as human-generated texts, even involving some landmarks (*e.g.*, sofa) that are informative yet missed in human reference. Fig. 3 (c) shows that, LANA can offer real-time behavioral interpretation by showing human text report of its navigation process. This not only eases human from consistent monitoring, but also reveals its inner mode to some extent. For example, the report at step 1-3 informs that LANA wrongly recognizes the storeroom as the bedroom – this is why LANA chooses to enter the storeroom at step 2. In short, as a language-capable navigator, LANA shows advantages in (post-hoc) interpretability and human-robot bi-directional communication, which are the basic premises of human trust generated.

5. Conclusion and Discussion

This work calls for a paradigm shift from current VLN agents – strong language-aided wayfinders but without language generation ability – towards more language-capable navigation robots that can not only execute navigation instructions but also verbally describe the navigation routes. We present LANA, which learns to master both instruction following and generation with one single model. LANA performs on par or even better than previous task-specific solutions in both tasks, with much reduced complexity. Crucially, LANA can write high-quality route descriptions that are informative to interpret its behavior and direct humans in collaboration. We believe LANA provides a solid basis for the creation of language-capable robots and brings us closer to the ultimate goal of building socially-intelligent and trustworthy robots. Future work should reinforce LANA with the knowledge of large-scale pretrained foundation models.

Acknowledge This work is supported by National Key R&D Program of China (No. 2020AAA0108800) and the Fundamental Research Funds for the Central Universities (No. 226-2022-00051).

References

[1] Sanyam Agarwal, Devi Parikh, Dhruv Batra, Peter Anderson, and Stefan Lee. Visual landmark selection for generating grounded and interpretable navigation instructions. In *CVPR Workshop*, 2019. 1, 2, 6

[2] Gary L Allen. From knowledge to words to wayfinding: Issues in the production and comprehension of route directions. In *International Conference on Spatial Information Theory*, 1997. 2

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 6

[4] 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 *CVPR*, 2018. 1, 2, 5, 6, 7

[5] Jacob Andreas and Dan Klein. Alignment-based compositional semantics for instruction following. In *EMNLP*, 2015. 2

[6] Sean Andrist, Erin Spannan, and Bilge Mutlu. Rhetorical robots: making robots more effective speakers using linguistic cues of expertise. In *International Conference on Human-Robot Interaction*, 2013. 2

[7] Jacob Arkin, Daehyung Park, Subhro Roy, Matthew R Walter, Nicholas Roy, Thomas M Howard, and Rohan Paul. Multimodal estimation and communication of latent semantic knowledge for robust execution of robot instructions. *The International Journal of Robotics Research*, 39(10-11):1279–1304, 2020. 2

[8] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, 2005. 6

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 5

[10] David L Chen and Raymond J Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI*, 2011. 2

[11] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021. 2, 3, 4, 5, 6

[12] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Learning from unlabeled 3d environments for vision-and-language navigation. In *ECCV*, 2022. 2, 3

[13] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, 2022. 2

[14] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020. 3

[15] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 5

[16] Heriberto Cuayáhuitl, Nina Dethlefs, Lutz Frommberger, Kai-Florian Richter, and John Bateman. Generating adaptive route instructions using hierarchical reinforcement learning. In *International Conference on Spatial Cognition*, 2010. 2

[17] Amanda Cercas Curry, Dimitra Gkatzia, and Verena Rieser. Generating and evaluating landmark-based navigation instructions in virtual environments. In *European Workshop on Natural Language Generation*, 2015. 2

[18] Robert Dale, Sabine Geldof, and J Prost. Using natural language generation in automatic route. *Journal of Research and Practice in Information Technology*, 36(3):23, 2004. 2

[19] Andrea F Daniele, Mohit Bansal, and Matthew R Walter. Navigational instruction generation as inverse reinforcement learning with neural machine translation. In *International Conference on Human-Robot Interaction*, 2017. 1, 2

[20] Laure De Cock, Kristien Ooms, Nico Van de Weghe, Nina Vanhaeren, Pieter Pauwels, and Philippe De Maeyer. Identifying what constitutes complexity perception of decision points during indoor route guidance. *International Journal of Geographical Information Science*, 35(6):1232–1250, 2021. 2

[21] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. In *NeurIPS*, 2020. 2, 6

[22] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *CVPR*, 2021. 3

[23] Zi-Yi Dou and Nanyun Peng. Foam: A follower-aware speaker model for vision-and-language navigation. In *NAACL*, 2022. 1, 2

[24] Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. The role of trust in automation reliance. *International Journal of Human-Computer Studies*, 58(6):697–718, 2003. 2

[25] Gary W Evans, David G Marrero, and Patricia A Butler. Environmental learning and cognitive mapping. *Environment and Behavior*, 13(1):83–104, 1981. 2

[26] Daniel Fried, Jacob Andreas, and Dan Klein. Unified pragmatic models for generating and following instructions. *arXiv preprint arXiv:1711.04987*, 2017. 2

[27] 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 *NeurIPS*, 2018. 1, 2, 5, 6, 7

[28] Tsu-Jui Fu, Xin Wang, Matthew Peterson, Scott Grafton, Miguel Eckstein, and William Yang Wang. Counterfactual vision-and-language navigation via adversarial path sampling. In *ECCV*, 2020. 2

[29] Robert Goeddel and Edwin Olson. Dart: A particle-based method for generating easy-to-follow directions. In *International Conference on Intelligent Robots and Systems*, 2012. 2

[30] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *ICCV*, 2021. 2, 3, 5, 6

[31] 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 *CVPR*, 2020. 2, 3, 5, 6

[32] Yicong Hong, Cristian Rodriguez-Opazo, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. In *NeurIPS*, 2020. 5

[33] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. A recurrent vision-and-language bert for navigation. In *CVPR*, 2021. 2, 3, 5, 6

[34] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *CVPR*, 2022. 5

[35] Haoshuo Huang, Vihan Jain, Harsh Mehta, Alexander Ku, Gabriel Magalhaes, Jason Baldridge, and Eugene Ie. Transferable representation learning in vision-and-language navigation. In *ICCV*, 2019. 2

[36] Zanming Huang, Zhongkai Shangguan, Jimuyang Zhang, Gilad Bar, Matthew Boyd, and Eshed Ohn-Bar. Assister: Assistive navigation via conditional instruction generation. In *ECCV*, 2022. 2

[37] Alycia M Hund and Jennifer L Minarik. Getting from here to there: Spatial anxiety, wayfinding strategies, direction type, and wayfinding efficiency. *Spatial Cognition and Computation*, 6(3):179–201, 2006. 2

[38] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *ACL*, 2019. 2, 5, 6, 7

[39] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *CVPR*, 2019. 2

[40] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 3

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[42] Benjamin Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2(2):129–153, 1978. 2

[43] Federico Landi, Lorenzo Baraldi, Marcella Cornia, Massimiliano Corsini, and Rita Cucchiara. Perceive, transform, and act: Multi-modal attention networks for vision-and-language navigation. *arXiv preprint arXiv:1911.12377*, 2020. 6

[44] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 3

[45] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah A Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. In *EMNLP*, 2019. 3

[46] Chen Liang, Wenguan Wang, Tianfei Zhou, Jiaxu Miao, Yawei Luo, and Yi Yang. Local-global context aware transformer for language-guided video. *PAMI*, 2023. 3

[47] Chen Liang, Wenguan Wang, Tianfei Zhou, and Yi Yang. Visual abductive reasoning. In *CVPR*, 2022. 3

[48] Chuang Lin, Yi Jiang, Jianfei Cai, Lizhen Qu, Gholamreza Haffari, and Zehuan Yuan. Multimodal transformer with variable-length memory for vision-and-language navigation. In *ECCV*, 2022. 2

[49] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004. 6

[50] Gary Look, Buddhika Kottahachchi, Robert Laddaga, and Howard Shrobe. A location representation for generating descriptive walking directions. In *Proceedings of the International Conference on Intelligent User Interfaces*, 2005. 2

[51] Kristin L Lovelace, Mary Hegarty, and Daniel R Montello. Elements of good route directions in familiar and unfamiliar environments. In *International Conference on Spatial Information Theory*, 1999. 2

[52] Kevin Lynch. *The Image of the City*. The MIT Press, 1960. 2

[53] 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 *ICLR*, 2019. 2

[54] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *CVPR*, 2019. 2

[55] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: connecting language, knowledge, and action in route instructions. In *AAAI*, 2006. 2

[56] 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. In *ECCV*, 2020. 2, 5

[57] Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *AAAI*, 2016. 2

[58] Stefan Oßwald, Henrik Kretzschmar, Wolfram Burgard, and Cyrill Stachniss. Learning to give route directions from human demonstrations. In *ICRA*, 2014. 1, 2

[59] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6

[60] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *ICCV*, 2021. 2

[61] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In *ECCV*, 2020. 6

[62] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, 2020. 2, 5, 6

[63] Yanyuan Qiao, Yuankai Qi, Yicong Hong, Zheng Yu, Peng Wang, and Qi Wu. Hop: History-and-order aware pre-training for vision-and-language navigation. In *CVPR*, 2022. 2, 3, 5, 6

[64] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[65] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsu-

pervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. 4

[66] Kai-Florian Richter and Matt Duckham. Simplest instructions: Finding easy-to-describe routes for navigation. In *International Conference on Geographic Information Science*, 2008. 2

[67] Raphael Schumann and Stefan Riezler. Generating landmark navigation instructions from maps as a graph-to-text problem. In *ACL*, 2021. 2

[68] Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariët Theune. Report on the second challenge on generating instructions in virtual environments (give-2.5). In *European Workshop on Natural Language Generation*, 2011. 2

[69] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 3

[70] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL*, 2019. 1, 2, 6, 7

[71] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*, 2011. 1, 2

[72] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, 2020. 1, 2

[73] Eric J Vanetti and Gary L Allen. Communicating environmental knowledge: The impact of verbal and spatial abilities on the production and comprehension of route directions. *Environment and Behavior*, 20(6):667–682, 1988. 2

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[75] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 6

[76] David Waller and Yvonne Lippa. Landmarks as beacons and associative cues: their role in route learning. *Memory & Cognition*, 35(5):910–924, 2007. 2

[77] Hanqing Wang, Wei Liang, Jianbing Shen, Luc Van Gool, and Wenguan Wang. Counterfactual cycle-consistent learning for instruction following and generation in vision-language navigation. In *CVPR*, 2022. 1, 2, 3, 5, 6, 7

[78] Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. Towards versatile embodied navigation. In *NeurIPS*, 2022. 2

[79] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *CVPR*, 2021. 2

[80] Hanqing Wang, Wenguan Wang, Tianmin Shu, Wei Liang, and Jianbing Shen. Active visual information gathering for vision-language navigation. In *ECCV*, 2020. 2, 6

[81] Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. Less is more: Generating grounded navigation instructions from landmarks. In *CVPR*, 2022. 1

[82] 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 *CVPR*, 2019. 2, 5, 6

[83] 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 *ECCV*, 2018. 2

[84] Xiaohan Wang, Linchao Zhu, and Yi Yang. T2vlad: global-local sequence alignment for text-video retrieval. In *CVPR*, 2021. 3

[85] Xiaohan Wang, Linchao Zhu, Zhedong Zheng, Mingliang Xu, and Yi Yang. Align and tell: Boosting text-video retrieval with local alignment and fine-grained supervision. *IEEE TMM*, 2022. 3

[86] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. In *ICLR*, 2021. 3

[87] Shawn L Ward, Nora Newcombe, and Willis F Overton. Turn left at the church, or three miles north: A study of direction giving and sex differences. *Environment and Behavior*, 18(2):192–213, 1986. 2

[88] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1989. 5

[89] Wenhao Wu, Zhun Sun, and Wanli Ouyang. Transferring textual knowledge for visual recognition. In *AAAI*, 2023. 3

[90] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In *CVPR*, 2023. 3

[91] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 3

[92] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Centerclip: Token clustering for efficient text-video retrieval. In *SIGIR*, 2022. 3

[93] Yusheng Zhao, Jinyu Chen, Chen Gao, Wenguan Wang, Lirong Yang, Haibing Ren, Huaxia Xia, and Si Liu. Target-driven structured transformer planner for vision-language navigation. In *ACMMM*, 2022. 2, 3, 4

[94] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, 2020. 2, 3, 6