Counterfactual Cycle-Consistent Learning for Instruction Following and Generation in Vision-Language Navigation

Hanqing Wang[1,2], Wei Liang[1], Jianbing Shen[3], Luc Van Gool[2], Wenguan Wang[4*]

[1] Beijing Institute of Technology [2] ETH Zurich [3] SKL-IOTSC, University of Macau [4] ReLER, AAII, University of Technology Sydney

https://github.com/HanqingWangAI/CCC-VLN

Abstract

*Since the rise of vision-language navigation (VLN), great progress has been made in **instruction following** – building a follower to navigate environments under the guidance of instructions. However, far less attention has been paid to the inverse task: **instruction generation** – learning a speaker to generate grounded descriptions for navigation routes. Existing VLN methods train a speaker independently and often treat it as a data augmentation tool to strengthen the follower, while ignoring rich cross-task relations. Here we describe an approach that learns the two tasks simultaneously and exploits their intrinsic correlations to boost the training of each: the follower judges whether the speaker-created instruction explains the original navigation route correctly, and vice versa. Without the need of aligned instruction-path pairs, such cycle-consistent learning scheme is complementary to task-specific training targets defined on labeled data, and can also be applied over unlabeled paths (sampled without paired instructions). Another agent, called creator is added to generate counterfactual environments. It greatly changes current scenes yet leaves novel items – which are vital for the execution of original instructions – unchanged. Thus more informative training scenes are synthesized and the three agents compose a powerful VLN learning system. Extensive experiments on a standard benchmark show that our approach improves the performance of various follower models and produces accurate navigation instructions.*

1. Introduction

Vision-language navigation (VLN) [7], *i.e.*, enabling an agent to navigate across realistic environments given human instructions, has received great attention (Fig. 1(a)). Many powerful wayfinding agents (*i.e.*, *follower*) were developed to perform such embodied **instruction following** task. Unfortunately, the inverse task – learning a *speaker* that vividly explains navigation paths – has remained under-explored. In fact, **instruction generation** is also a crucial ability of AI agents. In many scenarios, AI agents should be able to communicate with humans for efficient collaboration, instead of executing instructions only [76, 29]. For example, when a human-robot team is doing search and rescue [68, 18, 75], the human may first issue commands (*e.g.*, "explore along this direction until the end of the hallway") that direct the robot to navigate a building and search for survivors. In this process, the robot is expected to report its progress (*e.g.*, "I have inspected three rooms") and explain its plan (*e.g.*, "I will continue to navigate this direction and stop at the end of the hallway") [69]. Robot's ability to generate linguistic explanations can help human to resolve potential ambiguity (*e.g.*, identifying "this direction" and "hallway") [18] and establish trust [9, 20]. Hence, the robot can even in turn help the human to navigate its explored area (which is unfamiliar to the human) [27], *e.g.*, "go straight and you will pass through four rooms before reaching the end of the hallway".

In addition to emphasizing the importance of instruction generation, we explore the intrinsic correlation between

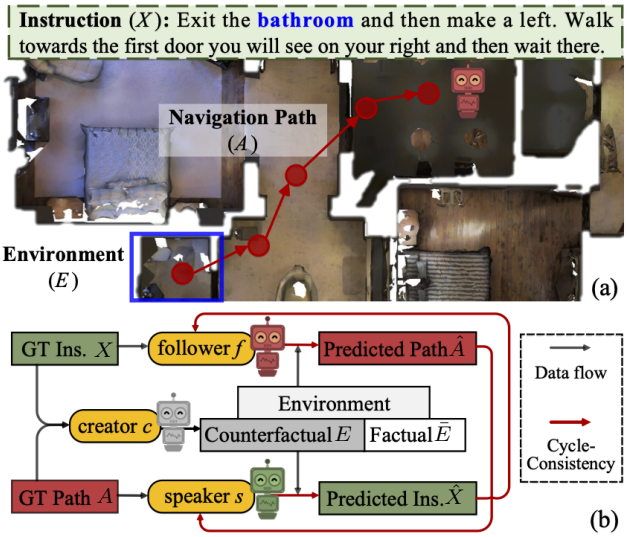

Figure 1: (a) VLN [7]. (b) Our counterfactual cycle-consistent (CCC) learning system consists of three agents, *i.e.*, speaker s for instruction generation, follower f for instruction following, and creator c for counterfactual environment creation.

*Corresponding author: *Wenguan Wang*.

instruction following and generation to derive a powerful VLN learning framework. Specifically, given the visual environment space $\mathcal{E}$, linguistic instruction space $\mathcal{X}$, and navigation path space $\mathcal{A}$, instruction following learns a follower $f: \mathcal{E} \times \mathcal{X} \mapsto \mathcal{A}$ that maps visual observations and navigable directions into action sequences, while instruction generation learns a speaker $s: \mathcal{E} \times \mathcal{A} \mapsto \mathcal{X}$ that maps observations and action sequences into trustable instructions. Clearly, there exists strong dependencies among the input and output spaces of s and f. Surprisingly, such task correlation was long ignored; current VLN methods only learn an isolated speaker as an one-off plugin for data augmentation [24, 66].

We instead propose to jointly train the speaker and follower in a compact, cycle-consistent learning framework (Fig. 1(b)). During training, $(E, X) \in \mathcal{E} \times \mathcal{X}$ is first mapped to $\hat{A} \in \mathcal{A}$ through the follower f (🤖) and then translated to an instruction $\tilde{X} \in \mathcal{X}$ through the speaker (🤖), *i.e.*, $s(E, \hat{A})$. In environment E, the dissimilarity between X and $\tilde{X}$, denoted as $\triangle_E(X, \tilde{X})$, is used as the feedback signal to regularize training. Similarly, given $(E, A) \in \mathcal{E} \times \mathcal{A}$, $\triangle_E(A, \tilde{A})$ can be estimated and used for training. As $\triangle$ errors are only about the cycle-consistency over (E, X) and (E, A), any other training objectives defined on labeled triplets, *i.e.*, (E, X, A), are compatible. Hence, we can apply such learning system on "unlabeled" data (E, A'), *i.e.*, sampling a path A' in an environment E without corresponding instruction. Thus both labeled instruction-path samples and unlabeled paths can be simultaneously used during training. This is more elegant than current *de facto* VLN training protocol [24, 66] that has three phases: i) train the follower and speaker separately on aligned instruction-path samples; ii) use the speaker to create synthetic instructions for randomly sampled paths; and iii) fine-tune the follower on the pseudo instruction-path samples. Further, as learning from pseudo-parallel data inevitably accompanies with the data quality problem (*i.e.*, the quality of the pseudo instruction-path samples is difficult to guarantee) [47], our speaker-follower collaborative learning game is more favored, *i.e.*, $\triangle$ errors can be viewed as quality scores and play as supervisory signals to boost the training of both f and s.

Besides the speaker and follower, another agent, called *creator* (🤖), is put into our cycle-consistent learning game. The creator serves as a plug-and-play component for **counterfactual environment synthesis**, enabling more robust training. Thinking about alternative possibilities for past or future events is central to human thinking [62]. We frequently construct counterfactuals ("counter to the facts"): what might happen if $\cdots$? Counterfactual thinking gives us the flexibility in learning from past limited experience through mental simulation. Nevertheless, this issue is rarely addressed in VLN. Recently, [25] interpreted current prevailing data augmentation technique [24] – back-translation – as a kind of counterfactual thoughts – given an instruction like"walk away from the door", the agent builds a counterfactual like "if I walk in the door, what should the instruction be?" by augmenting sampled paths with artificial instructions. This allows the agent to be more efficiently trained by performing alternative actions that it did not actually make [25]. Although our speaker-follower game naturally supports such path sampling based counterfactual thinking (*i.e.*, involving unlabeled data (E, A') during training, as mentioned before), our creator can build another kind of counterfactuals, *e.g.*, "if I change current environment by, for example, removing or putting in a lot of furniture that are irrelevant to the instruction, I will still execute the original navigation plan". Concretely, given an environment-instruction-path tuple $(E, X, A) \in \mathcal{E} \times \mathcal{X} \times \mathcal{A}$, the creator $c: \mathcal{E} \times \mathcal{X} \times \mathcal{A} \mapsto \mathcal{E}$ "images" a new environment $\bar{E} \in \mathcal{E}$, such that i) $\bar{E}$ is greatly different from E, and ii) $\bar{E}$ should be still aligned with the original instruction-path pair (X, A). With i) and ii), we address both diversity and realism, which are proved critical in counterfactual thinking [26, 84]. Hence, as $\bar{E}$ and (X, A) compose a new valid training sample, the cycle-consistent errors, *i.e.*, $\triangle_{\bar{E}}(A, \tilde{A})$ and $\triangle_{\bar{E}}(X, \tilde{X})$, can be estimated over $(\bar{E}, X, A)$. In this way, our speaker, follower and creator form a powerful learning system, which makes a clever use of cross-task and cross-modal connections as well as resembles a counterfactual thinking process.

Our counterfactual cycle-consistent (CCC) framework is optimized by policy gradient methods and compatible with current imitation learning (IL) and reinforcement learning (RL) based VLN training protocols. We apply our CCC over several VLN baseline models and test it on gold-standard R2R dataset [7]. Experimental results verify the efficacy of CCC on both instruction following and generation tasks.

2. Related Work

Vision-Language Navigation (Instruction Following). Although VLN [7] is a relatively new task in computer vision, its core part – instruction following [12, 53] – has been long studied in natural language processing and robotics. Early studies typically built the navigator in controlled environments with formulaic route descriptions [51, 68, 12, 8, 53, 54]. Anderson *et al*. thus introduced R2R dataset [7] to investigate embodied navigation in photo-realistic simulated environments [11] with human-created instructions. Soon after, numerous efforts were made towards: **i)** raising more efficient learning paradigms, *e.g.*, IL [7], hybrid of model-free and model-based RL [81], and ensemble of IL and RL [79]; **ii)** exploring extra supervisory signals from synthesized samples [24, 66, 25], auxiliary tasks [79, 37, 49, 90], or even massive web image-text paired data [52, 30]; **iii)** developing more powerful perception-linguistic embedding schemes [36, 60, 80, 35]; and **iv)** designing smarter path planning strategies by self-correction [38, 50], active exploration [74], or map building [73, 13, 19]. Some oth-

ers learn environment-agnostic representations [80], or focus on fine-grained instruction parsing [34].

Our work differs significantly from the above body of work. We address the importance of both instruction following and generation, instead of the navigation task only. The dual tasks form a closed loop for end-to-end joint training. We are particularly interested in how to leverage their intrinsic connections to better learn each other, within factual environments as well as counterfactual alternatives.

Instruction Generation. Though being less studied in computer vision [2], generating linguistic route instructions [16] has raised wide research interest in robotics [27], linguistics [65], cognition [40], and environmental psychology [70], and can be traced to Lynch's work [48] in 1960. Early efforts investigated the principles underlying the process of human constructing route descriptions [83, 4, 45] and the characteristics of "easy-to-follow" instructions [44, 72, 61]. They pointed out the importance of involving intuitive landmarks (*e.g.*, physical objects and locations) and concise topological descriptions (*e.g.*, turn-by-turn directions) in instructions [18]. Based on these studies, some simple systems [44, 27] for instruction creation were developed using handcrafted *templates*, *i.e.*, slotting the content into pre-built linguistic structures. Some complicated ones [17] made use of linguistically motivated *rules* or full-fledged *grammars*, to better emulate the way people compose instructions and produce outputs in a more flexible and extensible manner [22]. Recent solutions [15, 57, 18, 23] lean on end-to-end, data-driven techniques, without manually crafted templates or rules. But they are typically performed on simple grid or rendered environments and thus factor out the role of perception in instruction creation to some extent.

Instruction creation attains far less attention in VLN [2]. Only some data augmentation techniques [24, 66, 25] learn an instruction generator (speaker) with training pairs of aligned navigation paths and human instructions. Then they use the speaker to synthesize instructions for newly sampled paths as extra training examples. However, they only treat instruction creation as an auxiliary task and train the speaker and follower separately. Hence it is hard to control the quality of the pseudo instructions created by the speaker. Unlike these methods, we propose a unified framework that learns the speaker and follower simultaneously, and explicitly uses their correlation as a regularization term for robust training.

Cycle-Consistent Learning. Cycle-consistent learning explores task correlations to regularize training and can be implemented in different forms, such as forward-backward object tracking [77, 46], CycleGAN [91], and dual learning [31]. Taking dual learning as an example, its idea is intuitive: if we map an x from one domain to another and then map it back, we should recover the original x [89]. It was successfully applied to many tasks like neural machine translation [31], sentiment analysis [85], image-to-image translation [39], question answering [67, 64, 42], *etc*.

In a broader sense, our study can be viewed as the first attempt that explores the duality of instruction generation and following in embodied navigation tasks. Both the two tasks are learnt in a dual-task learning framework, where their symmetric structures are explored as informative feedback signals for boosting each, even with unlabeled samples.

Counterfactual Thinking. Counterfactual thinking [62] (*i.e.*, the construction of mental alternatives to reality) is crucial to how people learn from experience and predict the future, and can influence different cognitive behaviors such as deduction, decision making, and problem solving [86, 21]. Recent studies proved that counterfactual thoughts can improve the explainability [33, 28], fairness [41], and robustness [78] of trained models. The use of counterfactual examples has also been explored in the context of visual question answering [3, 1, 14] and open set recognition [56, 87].

Fu *et al*. [25] revisit the idea of back-translation based data augmentation from a counterfactual thinking perspective: extra routes are adversarially selected, instead of randomly sampled [24], and translated into instructions for smarter data augmentation. Apart from sampling paths in real environments, we learn a creator to generate new visual scenes as more effective counterfactuals. Although counterfactual environment synthesis is also addressed in [59], it is achieved by introducing minimum interventions that cause the follower to change its output. In contrast, we seek to modify the factual environments to maximum, on the premise of ensuring the original instructions can be still executed. The combination of diversity and realism makes generated environments useful as training examples. Moreover, we learn the speaker and follower end-to-end collaboratively.

3. Methodology

We address two related tasks, *i.e.*, instruction following and generation, under R2R VLN setting [7]. For instruction following, a follower f is learnt to find a route A to the target location, specified by the instruction X, in a 3D environment E. For instruction generation, a speaker s is learnt to create a description X for route A in E. Here, s and f are jointly trained in an end-to-end cycle-consistent framework (*cf.* §3.1) and a creator c is further introduced to synthesize counterfactual environments for boosting training (*cf.* §3.2).

3.1. Cycle-Consistent Learning for Instruction Following and Generation

We learn the two tasks jointly (Fig. 2(a)): the speaker s and follower f act as an evaluator for each other. s is used to evaluate the quality of $\hat{A}$ generated by $f(E, X)$ and returns the feedback signal $\triangle_E(X, s(E, \hat{A}))$ to f, and vice versa.

Follower. The follower f is instantiated as a `Seq2Seq` model which computes a distribution $P(A|X; E)$ over route A (*i.e.*, a serious of actions $A = \{a_t\}_{t=1}^{T}$) given instruction

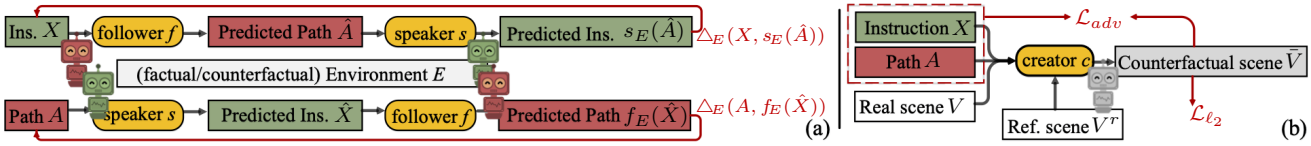

Figure 2: (a) Our cycle-consistent learning scheme (*cf.* §3.1) for the speaker s and follower f, trained over both factual and counterfactual environments. (b) Our creator c for counterfactual environment generation (*cf.* §3.2).

X (*i.e.*, a sequence of words $X=\{x_l\}_{l=1}^L$) under environment E. At each step t, the follower observes E as an image scene $V_t \subset E$. Conditioned on the visual and linguistic features, *i.e.*, $\boldsymbol{V}_t$ and $\boldsymbol{X}=\{\boldsymbol{x}_l\}_{l=1}^L$, and prior action embedding $\boldsymbol{a}_{t-1}$, the follower f first produces current hidden state $\boldsymbol{h}_t^f$:

$$\boldsymbol{h}_t^f = \text{LSTM}^f([\boldsymbol{V}_{t-1}, \boldsymbol{X}, \boldsymbol{a}_{t-1}], \boldsymbol{h}_{t-1}^f). \quad (1)$$

There are two basic follower designs in the literature, based on their definitions of the action space. The first-type followers [7] simplify the action space to six low-level *visuomotor* behaviors, *i.e.*, `left`, `right`, `up`, `down`, `forward` and `stop`. For example, `left` refers to turning left by 30°. The action embeddings are linguistic features and only a front view is perceived as V_t. Given proceeding actions $a_{1:t-1}$, instruction X and past observations $V_{1:t-1}$, the conditional probability of current action a_t is computed as:

$$P(a_t|a_{1:t-1}, x_{1:L}, V_{1:t-1}) = \text{softmax}_{a_t}(\boldsymbol{W}_1 \boldsymbol{h}_t^f), \quad (2)$$

where $\boldsymbol{W}_1 \boldsymbol{h}_t \in \mathbb{R}^6$ gives a score vector over the six actions.

The second-type followers [24, 79] first "look around" and gain a *panoramic* view as V_t. Then V_t is divided into 36 subviews, *i.e.*, $V_t=\{V_{t,i}\}_{i=1}^{36}$, forming current action space. Thus each action a_t is associated with a subview V_{t,a_t}, and $\boldsymbol{a}_t \equiv \boldsymbol{V}_{t,a_t}$. Then, the likelihood of a_t is formulated as:

$$P(a_t|a_{1:t-1}, x_{1:L}, V_{1:t-1}) = \text{softmax}_{a_t}(\boldsymbol{a}_t^\top \boldsymbol{W}_2 \boldsymbol{h}_t). \quad (3)$$

As the percepts $V_{1:t-1}$ are completely determined by the navigation actions $a_{1:t-1}$ in all our cases, we can further specify $P(A|X;E)$ according to the probability chain rule: $P(A|X;E)=\prod_t P(a_t|a_{1:t-1}, x_{1:L}, V_{1:t-1})$. And for notational simplicity, we abbreviate the probability distribution $P(a_t|a_{1:t-1}, x_{1:L}, V_{1:t-1})$ over actions at step t as $p_t(a_t)$.

To fully examine the efficacy of our CCC framework, we experiment with different follower architectures [7, 24, 79].

Speaker. The speaker s is built as a recurrent neural network based encoder-decoder architecture which computes a distribution $P(X|A;E)$ over possible instruction X ($=\{x_l\}_{l=1}^L$) given route A in environment E.

The encoder first embeds the sequences of the actions $\{a_t\}_{t=1}^T$ and visual observations $\{V_t\}_{t=1}^T$ along the route, and generates hidden states $\boldsymbol{o}_{1:T}^s$ using an LSTM:

$$\boldsymbol{o}_t^s = \text{LSTM-Encoder}^s([\boldsymbol{V}_t, \boldsymbol{a}_t], \boldsymbol{o}_{t-1}^s). \quad (4)$$

Afterwards, the decoder computes the conditional probability of each target word x_l given its proceeding words $x_{1:l-1}$ as well as the input embedding $\boldsymbol{o}_T^s$:

$$P(x_l|x_{1:l-1}, a_{1:T}, V_{1:T}) = \text{LSTM-Decoder}^s(\boldsymbol{h}_l^s, \boldsymbol{x}_{l-1}, \boldsymbol{o}_T^s). \quad (5)$$

Finally, we have $P(X|A;E)=\prod_l p_l(x_l)$, where $p_l(x_l)=P(x_l|x_{1:l-1}, a_{1:T}, V_{1:T})$ for brevity. For fair comparison, our speaker is the same as, but not specific to, the one in [24].

Cycle-Consistent Training. Given aligned $(E,X) \in \mathcal{E}\times\mathcal{X}$, we can first obtain a navigation path $\hat{A}$ through the follower $f(E,X)$ (shortly $f_E(X)$). We then use the speaker to translate $\hat{A}$ to a visual-grounded instruction $s(E,\hat{A})$ (shortly $s_E(\hat{A})$), which is expected to be semantically similar to X, *i.e.*, gaining a small cycle-consistent error $\triangle_E(X, s_E(\hat{A}))$ (shortly $\triangle_E^X$). Similarly, for paired $(E,A) \in \mathcal{E}\times\mathcal{A}$, we have $\triangle_E^A$. Then the $\triangle$ error can be specified as the negative log-likelihood, which is minimized for regularizing training:

$$\begin{aligned}
\triangle_E^A &= -\log \sum_{\hat{X}\in\mathcal{X}} P(s_E(A)=\hat{X}|A) P(f_E(\hat{X})=A|s_E(A)=\hat{X}), \\
\triangle_E^X &= -\log \sum_{\hat{A}\in\mathcal{A}} P(f_E(X)=\hat{A}|X) P(s_E(\hat{A})=X|f_E(X)=\hat{A}).
\end{aligned} \quad (6)$$

For ease of reference, we briefly denote Eq. 6 as:

$$\begin{aligned}
\triangle_E^A &= -\log \sum\nolimits_{\hat{X}\in\mathcal{X}} P(\hat{X}|A;E) P(A|\hat{X};E), \\
\triangle_E^X &= -\log \sum\nolimits_{\hat{A}\in\mathcal{A}} P(\hat{A}|X;E) P(X|\hat{A};E).
\end{aligned} \quad (7)$$

Directly calculating the gradients for $\triangle_E^A$ is intractable due to the huge space of $\mathcal{X}$. A similar issue also holds for $\triangle_E^X$. Inspired by [31, 82], the gradients for $\triangle_E^A$ w.r.t. the parameters of the follower and speaker, *i.e.*, θ^f and θ^s, can be computed as (and similarly for $\triangle_E^X$):

$$\begin{aligned}
\frac{\partial \triangle_E^A}{\partial \theta^f} &\approx -\mathbb{E}_{\hat{X}\sim P(\cdot|A;E)}\left[\frac{\partial \log P(A|\hat{X};E)}{\partial \theta^f}\right] \\
&\approx -\frac{\partial \log P(A|\hat{X};E)}{\partial \theta^f}, \\
\frac{\partial \triangle_E^A}{\partial \theta^s} &\approx -\mathbb{E}_{\hat{X}\sim P(\cdot|A;E)}\left[\log P(A|\hat{X};E) \frac{\partial \log P(\hat{X}|A;E)}{\partial \theta^s}\right] \\
&\approx -(\log P(A|\hat{X};E) - b^f) \frac{\partial \log P(\hat{X}|A;E)}{\partial \theta^s},
\end{aligned} \quad (8)$$

where b^f is a baseline to reduce the training variance and estimated by the mean of previous $\log P(A|\hat{X};E)$. Please see our supplementary for more details.

Hence, as the estimation for the $\triangle$ error does not require any aligned instruction-trajectory pairs (X,A), Eq. 6 can be naturally applied over both labeled and unlabeled paths for training. Let us denote $\mathcal{D}=\{(E_n, X_n, A_n)\}_{n=1}^N$ as a labeled data collection which consists of N aligned environment-instruction-path tuples. As in conventions [24, 66, 38], we can also build an unlabeled data collection, $\mathcal{U}=\{(E_m, A'_m)\}_{m=1}^M$ through sampling some paths A'_m from

existing environments E_m (but without instruction annotations). Then our cycle-consistent learning loss is defined as:

$$\mathcal{L}_{cycle} = \frac{1}{N} \sum_{(E,X,A) \in \mathcal{D}} (\triangle_E^A + \triangle_E^X) + \frac{1}{M} \sum_{(E,A') \in \mathcal{U}} \triangle_E^{A'}. \quad (9)$$

Other learning objectives for instruction following and generation, defined over the labeled triplets (E, X, A), are also compatible and used in our training stage (*cf.* §3.3).

Remark. Our cycle-consistency design is driven by two beliefs. First, a desired VLN agent should be able to ground both navigation actions and linguistic cues together on the visual environments. Thus it is necessary to explore both navigation planning and instruction creation in a unified learning scheme, enabling the agent to better capture cross-modal and cross-task connections. Second, assuming the generated instruction $s_E(A)$ is a valid rephrasing of the original X, a robust follower should execute this rephrasing $s_E(A)$ with the same navigation plan as the original X. A similar conclusion also holds for the speaker.

3.2. Counterfactual Environment Creation

Back-translation based data augmentation [24] has become a common practice in VLN. The central idea is to translate sampled paths into artificial instructions and use these synthesized environment-instruction-path tuples to augment the labeled data $\mathcal{D}$. Aside from learning the speaker and follower in isolation, it involves *multiple* training stages (*cf.* §1). Contrarily, our cycle-consistent learning, conducted on both labeled $\mathcal{D}$ and unlabeled data $\mathcal{U}$, has a unified training objective (*cf.* Eq. 9). Interestingly, recent studies [25] suggested the advantage of such path sampling strategy [24] in counterfactual thinking: a path A', sampled in E, and its artificial instruction $\hat{X}'$ compose a counterfactual. Although our cycle-consistent learning scheme has naturally involved such kind of counterfactuals during the computation of $\triangle_E^{A'}$ in Eq. 9, to fully explore the potential of counterfactual thinking, we further propose an environment creator c that generates counterfactual observations by greatly changing house layouts but without interfering with the execution of original instructions.

Creator. Our goal is to learn a creator $c : (E, X, A) \mapsto \bar{E}$ that observes the environment E, instruction X, navigation path A and generates a counterfactual environment $\bar{E}$ such that i) the differences between E and $\bar{E}$ are large; and ii) $\bar{E}$ and (X, A) present high compatibility. Such design is based upon previous research that has proved that, i) human prefer large modifications (even introducing elements not present in the real experience), during counterfactual thinking [26]; and, ii) the imagined world should not be completely divorced form the reality (with proper changes) [84].

For ease of optimization, instead of only considering the aligned triplet $(E, X, A) \in \mathcal{D}$, the creator c additionally uses other real scene E^r (sampled from $\mathcal{D}$) as the reference. By mixing E with E^r, it creates a counterfactual environment $\bar{E}$, under the above-mentioned constraints: i) **diversity**: replacing elements in E, as many as possible, with the ones in E^r; ii) **realism**: maintaining (X, A) still feasible in the changed environment $\bar{E}$. To do so, a compact descriptor $\boldsymbol{u}$ is first generated for (E, X, A):

$$\boldsymbol{u} = \boldsymbol{h}_T^c, \quad \boldsymbol{h}_t^c = \text{LSTM}^c([\boldsymbol{V}_t, \boldsymbol{a}_t, \boldsymbol{X}], \boldsymbol{h}_{t-1}^c). \quad (10)$$

Here $\boldsymbol{u}$ is desired to encode all the necessary information that make (E, X, A) valid. As shown in Fig. 2(b), given an observed scene $V \in E$ and its reference $V^r \in E^r$, the creator c fuses them together, in the feature rather than pixel space:

$$\begin{aligned} \boldsymbol{q}_k &= \text{softmax}([\boldsymbol{v}_1^r, \boldsymbol{v}_2^r, \cdots, \boldsymbol{v}_K^r]^\top \cdot \boldsymbol{v}_k), \\ \boldsymbol{g}_k &= [\boldsymbol{v}_1^r, \boldsymbol{v}_2^r, \cdots, \boldsymbol{v}_K^r] \cdot \boldsymbol{q}_k, \\ \lambda_k &= \text{sigmoid}(\boldsymbol{u}^\top \boldsymbol{W}_3 \boldsymbol{v}_k), \\ \bar{\boldsymbol{v}}_k &= \lambda_k \boldsymbol{v}_k + (1 - \lambda_k) \boldsymbol{g}_k, \end{aligned} \quad (11)$$

where $\boldsymbol{v}_k$ is the embedding of a visual element v_k in scene V (*i.e.*, $\boldsymbol{V} = \text{max-pool}([\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_K])$), $\boldsymbol{q}_k$ refers to normalized correlation score vector between v_k and $V^r = \{v_k^r\}_k$, $\boldsymbol{g}_k$ indicates the attention summary, and λ_k gives the importance of $\boldsymbol{v}_k$ for the successful execution of (E, X, A) and decides whether $\boldsymbol{v}_k$ needs to be replaced. Eventually, we have $\bar{\boldsymbol{v}}_k$, *i.e.*, the embedding of a visual region $\bar{v}_k$ in the created counterfactual scene $\bar{V} \in \bar{E}$.

Training Objective. With the purposes of i) modifying the original scene V as much as possible and ii) keeping crucial information/landmarks aligned with the instruction-path pair (X, E), the training loss of the creator c is designed as:

$$\begin{aligned} \mathcal{L}^c &= \mathcal{L}_{\ell_2} + \mathcal{L}_{adv}, \\ &= ||\boldsymbol{\lambda}||_2 + \log(1 - d(\bar{V}, X, A)). \end{aligned} \quad (12)$$

The ℓ_2-norm loss $\mathcal{L}_{\ell_2}$ inspires the sparsity of $\boldsymbol{\lambda} = [\lambda_k]_k$ and hence addresses i). The adversarial loss $\mathcal{L}_{adv}$ tries to "fool" a discriminator $d : \mathcal{E} \times \mathcal{X} \times \mathcal{A} \mapsto [0, 1]$. The discriminator d is learnt to estimate the alignment between environments and instruction-action pairs by minimizing $d(\bar{V}, X, A) - d(V, X, A)$. Thus $\mathcal{L}_{adv}$ addresses ii). Note that d only uses geometric action embedding. Optimizing above objectives makes $(\bar{E}, X, A)$ a valid training example. Therefore, in our counterfactual environment $\bar{E}$, we can still train the speaker to translate the navigation path E into X, and train the follower to execute the instruction X as A.

Remark. The creator is fully differentiable and trained with the speaker and follower together, leading to a triple-agent learning system. During training, the cycle-consistent errors, *i.e.*, $\triangle_{\bar{E}}^A$ and $\triangle_{\bar{E}}^X$, can also be estimated and minimized over the counterfactual samples, *i.e.*, $(\bar{E}, X, A)$, generated by the creator. Moreover, the creator can also access the supervision signals of cycle-consistent learning (*cf.* Eq. 9) and training objectives for instruction following and generation learning (detailed in §3.3). Thus the creator can progressively create more informative counterfactuals on-the-fly, in turn boosting the training of the speaker and follower.

3.3. Implementation Details

Network Architecture. We implement our **follower** f with different architectures [7, 24, 79]. Instruction embedding $\boldsymbol{X}$ is from an LSTM based linguistic encoder as normally. With [7], only front view is used and corresponding embedding $\boldsymbol{V}$ is obtained from a pretrained ResNet-152 [32]. Action embedding $\boldsymbol{a}$ is also from the linguistic LSTM. With [24, 79], the panoramic view is perceived and divided into 36 sub-views (12 headings $\times$ 3 elevations with 30° intervals). Each sub-view is associated with a geometric feature, *i.e.*, $(\cos\phi_h, \sin\phi_h, \cos\phi_e, \sin\phi_e)$, where ϕ_h and ϕ_e are the angles of heading and elevation, respectively. Visual and geometric features are concatenated as the embedding of the sub-view and corresponding action. See [7, 24, 79] for more network details. The model design of our **speaker** s follows the one in [24], which is built upon the panoramic view system. For the **creator** c, the discriminator d only uses geometric information based action representation (to filter out the visual cues in trajectories). Both c and d adopt cross-modal co-attention based network architectures, like [66].

Training. Besides minimizing the cycle-consistent loss $\mathcal{L}_{cycle}$ (*cf.* Eq. 9), our CCC framework is also learnt with the training objectives for instruction generation and following, over the labeled data $\mathcal{D}$ and counterfactual samples. For instruction following, IL [7] is adopted for off-policy learning, where the loss is defined over the groundtruth navigation action sequence A: $\mathcal{L}^f_{IL} = -\log P(A|X)$. RL is also applied for on-policy learning [79, 66], *i.e.*, optimizing $\mathcal{L}^f_{RL} = -\sum_t \log p_t(a_t)\Lambda_t$, where $a_t \sim p_t(a_t)$ and Λ_t indicates the advantage in A2C [55]. For instruction generation, the speaker is trained with $\mathcal{L}^s = -\log P(X|A)$, where X refers to the groundtruth navigation instruction. To stabilize training, we apply an annealing strategy [66] to the IL signal which makes the agents learn a good initial policy.

Inference. Once trained, the speaker and follower can perform their specific tasks independently. As in conventions, we apply greedy prediction, *i.e.*, $x^*_l = \arg\max(p_l(x_l))$ and $a^*_t = \arg\max(p_t(a_t))$, for instruction creation and following, as approximations of $X^* = \arg\max P(X|A;E)$ and $A^* = \arg\max P(A|X;E)$, respectively.

Model	val seen				val unseen				test unseen			
	SR↑	NE↓	OR↑	SPL↑	**SR**↑	NE↓	OR↑	SPL↑	**SR**↑	NE↓	OR↑	SPL↑
Seq2Seq [7]	39.4	6.0	51.7	33.8	22.1	7.8	27.7	19.1	20.4	7.9	26.6	18.0
+ BT [24]	43.7	5.3	58.1	37.2	22.6	7.7	28.9	19.9	21.0	7.8	26.2	18.8
+ APS [25]	48.2	**5.0**	60.8	40.1	24.2	7.1	32.7	20.4	22.5	**7.5**	30.1	19.3
+ CCC	**50.1**	**5.0**	**61.1**	**42.6**	**28.4**	**6.8**	**35.3**	**22.1**	**25.5**	7.8	**35.9**	**20.6**
Speaker-Follower [24]	51.7	5.0	61.6	44.4	29.9	6.9	40.7	21.0	30.9	7.0	41.2	24.0
+ BT [24]	66.4	3.7	74.2	59.8	36.1	6.6	46.6	28.8	34.8	6.6	43.4	29.2
+ APS [25]	68.2	**3.3**	**74.9**	**62.5**	38.8	6.1	46.7	32.1	36.1	6.5	44.2	28.8
+ CCC	**68.4**	**3.3**	74.5	61.4	**43.5**	**5.8**	**52.0**	**38.1**	**41.4**	**5.9**	**51.0**	**36.6**
RCM [79]	47.0	5.7	53.8	44.3	35.0	6.8	43.0	31.4	35.9	6.7	43.5	33.1
+ BT [24]	61.9	4.1	66.9	58.6	45.6	5.7	52.4	41.8	44.5	5.9	52.4	40.8
+ APS [25]	63.2	3.9	69.3	59.5	47.7	5.4	56.6	42.8	45.1	5.8	53.9	40.9
+ CCC	**68.0**	**3.4**	**77.5**	**62.1**	**50.4**	**5.2**	**57.8**	**46.4**	**51.0**	**5.3**	**57.2**	**48.2**

Table 1: Quantitative comparison results (§4.1) for instruction following on R2R dataset [7].

4. Experiment

4.1. Performance on Instruction Following

Dataset. We conduct experiments on R2R [7], originally developed for the instruction following task. R2R has four sets: `train` (61 environments, 14,039 instructions), `val seen` (61 environments, 1,021 instructions), `val unseen` (11 environments, 2,349 instructions), and `test unseen` (18 environments, 4,173 instructions). There are no overlapping environments between the unseen and training sets.

Evaluation Metric. Following [7, 24], four standard metrics for instruction following are used: 1) *Success rate* (SR) computes the percentage of final positions less than 3 m away from the goal location. 2) *Navigation error* (NE) refers to the shortest distance between agent's final position and the goal location. 3) *Oracle success rate* (OR) is the success rate if the agent can stop at the closest point to the goal along its trajectory. 4) *Success rate weighted by path length* (SPL) [5] is a trade-off between SR and navigation length.

Evaluation Protocol. As in [24, 25], we test our model with several representative baselines [7, 24, 79] using different architectures, action spaces, and learning paradigms.

- *Seq2Seq* [7]: an attention-based `Seq2Seq` model that is trained with IL under the visuomotor action space.
- *Speaker-Follower* [24]: a compositional model that is trained with IL under the panoramic action space.
- *RCM* [79]: an improved multi-modal model that is trained using both IL and RL under the panoramic action space.

The baselines are trained with labeled instruction-path pairs in R2R `train` set. For each baseline, we further report the performance with our CCC and other two speaker-based data augmentation techniques, *i.e.*, back-translation (BT) [24] and adversarial path sampling (APS) [25]:

- CCC: our speaker is jointly learnt with the follower in real and counterfactual environments and translates randomly sampled paths into instructions as extra training data.
- BT [24]: the speaker is trained in isolation with the follower in real environments only and translates randomly sampled paths into instructions as extra training data.
- APS [25]: the speaker is trained in isolation with the follower in real environments only but translates adversarially selected paths into instructions as extra training data.

Quantitative Result. The comparison results on instruction following are summarized in Table 1. We find CCC outperforms other learning paradigms across diverse dataset splits and metrics. For the three baseline followers [7, 24, 79], CCC gains remarkable SR improvements (*i.e.*, 0.2-4.8,

Models	test unseen			
	SR↑	NE↓	OR↑	SPL↑
Self-Monitoring [49]	43.0	6.0	55.0	32.0
Regretful [50]	48.0	5.7	56.0	40.0
OAAM [60]	53.0	-	61.0	50.0
Tactical Rewind [38]	54.0	5.1	64.0	41.0
AuxRN [90]	55.0	5.2	62.0	**51.0**
E-Dropout [66]	48.0	5.6	58.0	44.0
E-Dropout [66] + CCC	52.2	5.1	59.8	46.9
Active Perception [74]	55.7	4.8	**73.1**	37.1
Active Perception [74] + CCC	60.6	4.3	71.4	41.3
SSM [73]	57.3	4.7	68.2	44.1
SSM [73] + CCC	**62.2**	**4.3**	72.3	49.2

Table 2: Benchmarking results (§4.1) for instruction following on R2R dataset [7].

2.7-4.7, and 3.0-6.1) compared with the second best on `val seen`, `val unseen` and `test unseen` respectively. This validates the efficacy of CCC across different follower architectures. Further, the performance improvements on `unseen` sets are relatively more significant, showing that CCC strengthens models' generalization ability.

Performance Benchmarking. For comprehensive evaluation, we conduct performance benchmarking by applying our CCC technique to [66, 73, 74], which are current top-leading instruction followers with public implementations:

- *E-Dropout* [66]: a multi-modal model that uses 'environmental dropout' method to mimic unseen environments.
- *Active Perception* [74]: a robust model that is able to actively explore surroundings for more intelligent planning.
- *SSM* [73]: a graph model which is equipped with a map building module for global decision making.

As shown in Table 2, our CCC greatly boosts the performance of current three top-performing instruction followers [66, 73, 74], across all the metrics. For example, SR and SPL of E-Dropout [66] are improved by 4.2 and 2.9, respectively. For Active Perception [74], it obtains significant performance gains, *e.g.*, 4.9 SR and 4.2 SPL. Based on SSM [73], our CCC further improves the SR to 62.2.

4.2. Performance on Instruction Generation

Dataset. As R2R [7] `test unseen` set is preserved for benchmarking instruction following methods, we report the performance of instruction generation on `val seen` and `val unseen` sets. Note that, in R2R, each path is associated with three ground-truth navigation instructions.

Evaluation Metric. Five standard textual evaluation metrics are considered here [2]: 1) BLEU [58] refers to the geometric mean of n-gram precision scores computed over reference and candidate descriptions. 2) CIDEr [71] first represents each sentence with a set of 1-4 grams and calculates the co-occurrences of n-grams in the reference sentences and candidate sentence as the score. 3) METEOR [10] is defined as the harmonic mean of precision and recall of unigram matches between sentences. 4) ROUGE [43] is computed by comparing overlapping n-grams, word sequences and word pairs. 5) SPICE [6] is based on the agreement of the scenegraph [63] tuples of the candidate sentence and all reference sentences. These metrics are calculated by comparing each candidate instruction to the three reference instructions given the navigation path. As suggested by [88], SPICE is adopted as the primary metric.

Evaluation Protocol. As we implement our follower over different baselines (*i.e.*, *Seq2Seq*[7], *Speaker-Follower*[24], and *RCM* [24]), we report the performance of their corresponding speakers (*i.e.*, Ours-*Seq2Seq*, Ours-*Speaker-Follower*, and Ours-*RCM*). For comparative methods, we consider the speaker in [24] (*i.e.*, BT Speaker) and the instruction generation model (*i.e.*, VLS) in [2]. The former is widely used in current VLN instruction following methods for data augmentation, and VLS is the only model that is specifically designed for instruction generation in VLN.

Quantitative Result. The comparison results on instruction generation are reported in Table 3. As seen, our speakers achieve better performance on most of the metrics. For example, our three speakers gain the top three SPICE scores on both `val seen` and `val unseen` sets. Moreover, among our three speakers, Ours-*RCM* performs best. We suspect that a stronger follower can provide better feedback signals to the speaker during cycle-consistent learning.

User Study. We organize two user studies. In the first user study, we sample 500 paths in `val unseen` and generate instructions through our three speakers (*i.e.*, Ours-*Seq2Seq*, Ours-*Speaker-Follower*, Ours-*RCM*). 25 volunteer students are asked to select the most meaningful instruction from those compared for each path. Ours-*RCM* receives more votes than the others (Ours-*RCM*: 40.5% *vs* Ours-*Speaker-Follower*: 32.1% *vs* Ours-*Seq2Seq*: 27.4%). In the second user study, we let other 25 students compare the outputs of Ours-*RCM*, BT Speaker and VLS. Ours-*RCM* wins with 68.6% picking rate (BT Speaker: 13.2%, VLS: 18.2%).

Model	val seen						val unseen					
	Bleu-1 ↑	Bleu-4 ↑	CIDEr ↑	Meteor ↑	Rouge ↑	**SPICE** ↑	Bleu-1 ↑	Bleu-4 ↑	CIDEr ↑	Meteor ↑	Rouge ↑	**SPICE** ↑
BT Speaker [24]	0.537	0.155	0.121	0.233	0.350	0.203	0.522	0.142	0.114	0.228	0.346	0.188
VLS [2]	0.549	0.157	0.137	0.228	0.352	0.214	0.548	0.159	0.132	0.231	0.357	0.197
Ours-*Seq2Seq*	0.720	0.296	0.529	0.233	0.487	0.216	0.704	0.273	0.475	0.229	0.473	0.202
Ours-*Speaker-Follower*	0.723	**0.299**	**0.566**	0.235	0.490	0.229	0.706	**0.275**	**0.477**	0.229	0.474	0.207
Ours-*RCM*	**0.728**	0.287	0.543	**0.236**	**0.493**	**0.231**	**0.708**	0.272	0.461	**0.231**	**0.477**	**0.214**

Table 3: Quantitative comparison results (§4.2) for instruction generation on R2R dataset [7].

Model	Component	Instruction Following				Instruction Generation					
		SR↑	NE↓	OR↑	SPL↑	Bleu-1 ↑	Bleu-4 ↑	CIDEr ↑	Meteor ↑	Rouge ↑	**SPICE** ↑
Baseline [24]	-	29.9	6.9	40.7	21.0	0.522	0.142	0.114	0.228	0.346	0.188
Cycle-Consistency	$\triangle_E^A$	33.2	6.7	44.7	23.7	0.694	0.269	0.446	0.228	0.471	0.192
	$\triangle_E^X$	28.6	6.9	40.1	20.5	0.699	0.271	0.456	0.228	0.472	0.195
	$\triangle_E^A+\triangle_E^X$	35.1	6.7	44.5	25.4	0.702	0.272	0.462	0.229	0.472	0.196
	$\triangle_E^A+\triangle_E^X+\triangle_E^{A'}$	37.7	6.6	45.9	26.7	0.703	0.272	0.467	0.229	0.471	0.198
Counterfactual Environment	*w/o* reference environment E^r	29.9	6.9	40.7	21.0	0.522	0.142	0.114	0.228	0.346	0.188
	w reference environment E^r	41.7	5.9	50.7	35.6	0.701	0.271	0.459	0.229	0.472	0.200
Full Model	$\triangle_E^A+\triangle_E^X+\triangle_E^{A'}+\triangle_{\tilde{E}}^A+\triangle_{\tilde{E}}^X$	43.5	5.8	52.0	38.1	0.706	0.275	0.477	0.229	0.474	0.207

Table 4: Ablation study on `val unseen` of R2R dataset [7]. See §4.3 for details.

Figure 3: Visual comparison results on instruction following (left) and instruction generation (right). See §4.3 for details.

4.3. Diagnostic Experiments

To fully examine the effectiveness of our core model designs, a set of ablative studies are conducted on `val unseen` of R2R [7] (Table 4). Our model is built upon [24].

Cycle-Consistent Learning. We first assess the contribution of our cycle-consistent learning scheme (*cf.* §3.1). For the baseline method, the follower and speaker are trained separately and independently. Then we build four alternatives, *i.e.*, $\triangle_E^A$, $\triangle_E^X$, $\triangle_E^A+\triangle_E^X$, and $\triangle_E^A+\triangle_E^X+\triangle_E^{A'}$. The first three baselines are trained by minimizing different $\triangle$ errors on the labeled dataset $\mathcal{D}$ only while the last one additionally uses unlabeled data $\mathcal{U}$, *i.e.*, 17K randomly sampled paths A' as in [24]. From Table 4, we can conclude that: i) leveraging cross-task relations indeed boosts the performance on both instruction following and generation (while $\triangle_E^X$ greatly facilitates instruction generation with small performance sacrifice of instruction following), and ii) our cycle-consistency learning scheme not only works well on labeled data, but also makes better use of unlabeled data.

Counterfactual Environment Creation. Next we study the efficacy of our counterfactual thinking strategy (*cf.* §3.2). To this end, we separately train the speaker and follower on the counterfactual examples created by the creator without cycle-consistent learning. We build a baseline '*w/o* reference environment E^r' by randomly masking a part of the original scene without considering reference environment. Table 4 shows that: i) our synthesized counterfactual environments can benefit the training of both the speaker and follower, and ii) additionally considering reference environments during synthesizing can produce more informative counterfactuals.

Full Model Design. After examining the contribution of two critical components individually, we evaluate the effectiveness of our full model design. Our full model learns the speaker, follower, and creator jointly with comprehensive use of labeled and unlabeled "real" data as well as counterfactual examples. As evidenced in Table 4, CCC brings more significant performance improvements over each individual module on both instruction following and generation.

Visual Comparison Results. Finally, some visual results are provided in Fig. 3 for more intuitive comparisons. From the left sub-figure we can observe that, the follower trained with our CCC framework can derive a more robust navigation policy and thus reaches the target location successfully. As shown in the right sub-figure, our speaker is able to generate more accurate instructions; some novel actions and landmarks are successfully mentioned. During training, our strong speaker in turn boosts the learning of the follower.

5. Conclusion

We introduced a powerful training framework, CCC, that learns navigation generation and following simultaneously. It explicitly leverages cross-task connections to regularize the training of both the speaker and follower. Hence, a creator is integrated into such cycle-consistent learning system, so as to synthesize counterfactual environments and further facilitate training. The CCC framework is model-agnostic and can be integrated into a diverse collection of navigation models, leading to performance improvements over both instruction following and generation, in the R2R dataset.

Acknowledgements This work was supported by Natural Science Foundation of China (NSFC) grant (No. 62172043) and ARC DECRA DE220101390.

References

[1] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *CVPR*, 2020. 3

[2] 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. 3, 7

[3] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal VQA: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *CVPR*, 2020. 3

[4] 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. 3

[5] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 6

[6] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 7

[7] 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, 3, 4, 6, 7, 8

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

[9] 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. 1

[10] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, 2005. 7

[11] 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. *3DV*, 2017. 2

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

[13] Kevin Chen, Junshen K. Chen, Jo Chuang, Marynel Vazquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *CVPR*, 2021. 2

[14] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*, 2020. 3

[15] 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. 3

[16] 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. 3

[17] 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. 3

[18] 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, 3

[19] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. In *NeurIPS*, 2021. 2

[20] 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. 1

[21] Kai Epstude and Neal J Roese. The functional theory of counterfactual thinking. *Personality and Social Psychology Review*, 12(2):168–192, 2008. 3

[22] Mary Ellen Foster. Natural language generation for social robotics: opportunities and challenges. *Philosophical Transactions of the Royal Society B*, 374(1771):20180027, 2019. 3

[23] Daniel Fried, Jacob Andreas, and Dan Klein. Unified pragmatic models for generating and following instructions. *arXiv preprint arXiv:1711.04987*, 2017. 3

[24] 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. 2, 3, 4, 5, 6, 7, 8

[25] 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, 3, 5, 6

[26] Vittorio Girotto, Donatella Ferrante, Stefania Pighin, and Michel Gonzalez. Postdecisional counterfactual thinking by actors and readers. *Psychological Science*, 2007. 2, 5

[27] 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. 1, 3

[28] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *ICML*, 2019. 3

[29] Scott A Green, Mark Billinghurst, XiaoQi Chen, and J Geoffrey Chase. Human-robot collaboration: A literature review and augmented reality approach in design. *International Journal of Advanced Robotic Systems*, 5(1):1, 2008. 1

[30] 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

[31] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *NeurIPS*, 2016. 3, 4

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.

Deep residual learning for image recognition. In *CVPR*, 2016. 6

[33] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *ECCV*, 2018. 3

[34] Yicong Hong, Cristian Rodriguez-Opazo, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. *NeurIPS*, 2020. 3

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

[36] Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. Are you looking? grounding to multiple modalities in vision-and-language navigation. In *ACL*, 2019. 2

[37] 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

[38] 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, 4, 7

[39] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 3

[40] Benjamin Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2(2):129–153, 1978. 3

[41] Matt Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *NeurIPS*, 2017. 3

[42] Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. Visual question generation as dual task of visual question answering. In *CVPR*, 2018. 3

[43] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004. 7

[44] 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. 3

[45] 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. 3

[46] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. Learning video object segmentation from unlabeled videos. In *CVPR*, 2020. 3

[47] Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*, 2019. 2

[48] Kevin Lynch. *The Image of the City*. The MIT Press, 1960. 3

[49] 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, 7

[50] 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, 7

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

[52] 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

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

[54] 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 *EMNLP*, 2018. 2

[55] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016. 6

[56] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *ECCV*, 2018. 3

[57] Stefan Oßwald, Henrik Kretzschmar, Wolfram Burgard, and Cyrill Stachniss. Learning to give route directions from human demonstrations. In *ICRA*, 2014. 3

[58] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 7

[59] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Qinfeng Shi, and Anton van den Hengel. Counterfactual vision-and-language navigation: Unravelling the unseen. In *NeurIPS*, 2020. 3

[60] 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. 2, 7

[61] Kai-Florian Richter and Matt Duckham. Simplest instructions: Finding easy-to-describe routes for navigation. In *International Conference on Geographic Information Science*, 2008. 3

[62] Neal J Roese. Counterfactual thinking. *Psychological Bulletin*, 121(1):133, 1997. 2, 3

[63] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *VL*, 2015. 7

[64] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *CVPR*, 2019. 3

[65] 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. 3

[66] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL*, 2019. 2, 3, 4, 6, 7

[67] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*, 2017. 3

[68] 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

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

[70] 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. 3

[71] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 7

[72] David Waller and Yvonne Lippa. Landmarks as beacons and associative cues: their role in route learning. *Memory & Cognition*, 35(5):910–924, 2007. 3

[73] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *CVPR*, 2021. 2, 7

[74] Hanqing Wang, Wenguan Wang, Tianmin Shu, Wei Liang, and Jianbing Shen. Active visual information gathering for vision-language navigation. In *ECCV*, 2020. 2, 7

[75] Haiyang Wang, Wenguan Wang, Xizhou Zhu, Jifeng Dai, and Liwei Wang. Collaborative visual navigation. *arXiv preprint arXiv:2107.01151*, 2021. 1

[76] Jijun Wang and Michael Lewis. Human control for cooperating robot teams. In *International Conference on Human-Robot Interaction*, 2007. 1

[77] Ning Wang, Yibing Song, Chao Ma, Wengang Zhou, Wei Liu, and Houqiang Li. Unsupervised deep tracking. In *CVPR*, 2019. 3

[78] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *CVPR*, 2020. 3

[79] 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, 4, 6

[80] Xin Wang, Vihan Jain, Eugene Ie, William Yang Wang, Zornitsa Kozareva, and Sujith Ravi. Environment-agnostic multitask learning for natural language grounded navigation. In *ECCV*, 2020. 2, 3

[81] 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

[82] Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. Multi-agent dual learning. In *ICLR*, 2019. 4

[83] 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. 3

[84] James Woodward. Psychological studies of causal and counterfactual reasoning. *Understanding Counterfactuals, Understanding Causation*, 2011. 2, 5

[85] Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. Dual supervised learning. In *ICML*, 2017. 3

[86] Yi Yang, Yueting Zhuang, and Yunhe Pan. Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. *Frontiers of Information Technology & Electronic Engineering*, 2021. 3

[87] Zhongqi Yue, Tan Wang, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Counterfactual zero-shot and open-set visual recognition. In *CVPR*, 2021. 3

[88] Ming Zhao, Peter Anderson, Vihan Jain, Su Wang, Alexander Ku, Jason Baldridge, and Eugene Ie. On the evaluation of vision-and-language navigation instructions. In *Conference of the European Chapter of the Association for Computational Linguistics*, 2021. 7

[89] Zhibing Zhao, Yingce Xia, Tao Qin, Lirong Xia, and Tie-Yan Liu. Dual learning: Theoretical study and an algorithmic extension. In *Asian Conference on Machine Learning*, 2020. 3

[90] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, 2020. 2, 7

[91] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 3