RES-StS: Referring Expression Speaker via Self-training with Scorer for Goal-Oriented Vision-Language Navigation

Liuyi Wang, Zongtao He, Ronghao Dang, Huiyi Chen, Chengju Liu, Qijun Chen, *Senior Member, IEEE*

***Abstract*—It is a rather practical but difficult task to find a specified target object via autonomous exploration based on natural language descriptions in an unstructured environment. Since the human-annotated data is expensive to gather for the goal-oriented vision-language navigation (GVLN) task, the size of the standard dataset is inadequate, which has significantly limited the accuracy of previous techniques. In this work, we aim to improve the robustness and generalization of the navigator by dynamically providing high-quality pseudo-instructions using a proposed RES-StS paradigm. Specifically, we establish a referring expression speaker (RES) to predict descriptive instructions for the given path to the goal object. Based on an environment-and-object fusion (EOF) module, RES derives spatial representations from the input trajectories, which are subsequently encoded by a number of transformer layers. Additionally, given that the quality of the pseudo labels is important for data augmentation while the limited dataset may also hinder RES learning, we propose to equip RES with a more effective generation ability by using the self-training approach. A trajectory-instruction matching scorer (TIMS) network based on contrastive learning is proposed to selectively use rehearsal of prior knowledge. Finally, all network modules in the system are integrated by suggesting a multi-stage training strategy, allowing them to assist one another and thus enhance performance on the GVLN task. Experimental results demonstrate the effectiveness of our approach. Compared with the SOTA methods, our method improves SR, SPL, and RGS by 4.72%, 2.55%, and 3.45% respectively, on the REVERIE dataset, and 4.58%, 3.75% and 3.14% respectively, on the SOON dataset.**

***Index Terms*—Goal-oriented Vision-Language Navigation, Self-training, Referring Expression Generation, Contrastive learning**

I. Introduction

As the instruction based on the referring expression like *"Please go to my office and bring me the document on the table"* are usually used in social conversations in our daily life, it is fundamental for embodied AI agents to have such a capability to accurately pursue the guidance and discern the goal object in perceptually-rich environments. The Goal-oriented Vision-Language Navigation (GVLN) setups in real environments [1, 2] take such a step toward this goal. Compared with the Vision-and-Language Navigation (VLN) task [3] that focuses on building a model capable of following fine-grained instructions, the GVLN task is more practical and constructive since fine-grained instructions are actually difficult to access in real life; thus the GVLN task has driven increased interest from various research fields. Although numerous methods have been explored to use visual and language clues to assist in goal navigation [1, 2, 4–7], the severe overfitting problem caused by the small dataset with monotonous environments still remains challenging. This problem will become prominent when the network scale significantly increases, resulting in weak generalization.

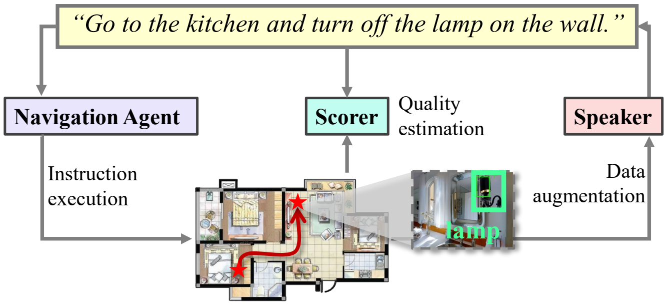


Fig. 1. Our method consists of three networks in total: a speaker, a scorer, and a navigation agent. We propose a RES-StS approach to generate high-quality pseudo labels for boosting the generalization and robustness of the navigation agent during training, where RES serves as a referring expression generation model, and self-training with TIMS is used to ensure the effectiveness of RES. Based on a multi-stage training strategy, these models jointly promote the performance of the GVLN task.

Since manually annotating instructions costs a lot, it is a potential direction to build an inverse model to dynamically generate instructions according to the easily obtained trajectories so that the dataset can be extended by in-domain pseudo pairs. The speaker-based model [8, 9] is such a potent method to serve as data augmentation in the VLN task. This inspires us to introduce the speaker to the GVLN task. However, since the speaker in the VLN task only needs to describe the actions for each step, it is different from the GVLN task that requires generating goal-oriented instructions. Therefore, the problem arises as to ***how to build a speaker model that can predict the referring expression with a description of the appearance, location, and surroundings of the target according to the***

L. Wang, Z. He, R. Dang, C. Liu and Q. Chen are with the Department of Control Science and Engineering, Tongji University, China. C. Liu is also with the Tongji Artificial Intelligence (Suzhou) Research Institute and Frontiers Science Center for Intelligent Autonomous Systems. H. Chen is with New York University, New York, USA. (e-mail: {wly, xingchen327, dangronghao, liuchengju, qjchen}@tongji.edu.cn; hc2446@nyu.edu) (*Corresponding authors: Chengju Liu, Qijun Chen*).

This paper is supported by the National Natural Science Foundation of China under Grants 61733013, 62173248, 62233013, and the Fundamental Research Funds for the Central Universities.

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3233554

given trajectory.

To address the above problem, we propose a referring expression speaker (RES) in this work, which is able to capture representations of visual trajectories with the goal object and predict descriptive natural language instructions. Specifically, we enrich the visual inputs by adding the object feature comprised of its visual features, position, and label derived by the trained object feature extractor. To fuse the abundant features and at the same time reduce unnecessary redundancy, we design an environment-and-object fusion (EOF) module. The EOF promotes the cross-fusion in the spatial domain by assigning the key and the query with different types of features based on the multi-head attention mechanism [10]. The goal object feature is concatenated with the trajectory features as an individual token to encourage the model to focus on the specific target representation. Moreover, motivated by the huge success made by the transformer with stronger long-term extraction ability, we choose the encoder-decoder transformer as the core of our network. Additionally, considering that the object category and the room type are two specific key points for the GVLN task, we develop a new metric named Hit Rate (HR) to assess the quality of generated instructions by determining the correct proportion of the dominant information present in the predicted instructions.

It should be highlighted that the speaker's performance is important because ineffective instruction creation might lead to poor navigational supervision. Although the speaker was primarily designed to produce pseudo labels for expanding the standard dataset, its limited scope may also make it difficult to train the speaker itself. Thus, another question follows: ***how to expand the knowledge of the speaker model beyond what it can acquire via supervised learning on the limited dataset?***

We suggest employing semi-supervised learning (SSL) [11–13] to improve training performance in order to get over this restriction. We train our RES model using the self-training framework [14–16], which first trains a teacher model on labeled data, predicts pseudo labels on unlabeled data, and then retrains a student model on the merged labeled and pseudo data. Next, even if SSL yields synthesize referring expression contexts via text prediction, there may still be noise due to the irrelevant description of the target object or incorrect localization to given trajectories. Therefore, we propose a trajectory-instruction matching scorer (TIMS) to utilize pseudo labels of high quality in a selective manner. TIMS is built based on the contrastive dual-encoder structure [17], which independently takes the completed trajectory and the instruction as inputs and computes the match similarity between these two modalities. We determine a data-driven threshold τ based on the average scores of positive pairs to filter out the low-quality pseudo labels. To justify the effectiveness of TIMS, we define a Matching Success Rate (MSR) to calculate the percentage of accurate matching predictions, and the model with the highest MSR will be chosen. We name the method of training RES via self-training with the scorer RES-StS.

As illustrated in Fig. 1, our approach involves a total of three networks: RES, TIMS, and the navigation agent (NA). Specifically, we choose two recent state-of-the-art models HAMT [6] and DUET [7] as the frameworks of our NA through experiments to verify the effectiveness of our proposed method. It is essential to reconcile these models properly for the goal of navigation and localization in the GVLN task. To this end, a multi-stage training strategy with four stages is proposed to integrate the whole training procedure. First, we train a teacher RES and TIMS on the small labeled dataset. Then, a large amount of unlabeled data with high-quality pseudo labels is predicted by the teacher RES based on the selection of TIMS. The mixed dataset is further used to retrain a student RES model and pre-train a NA model using several auxiliary tasks. Finally, we use the trained RES to dynamically replicate unseen environments with associated instructions for fine-tuning the NA model based on the back-translation with the environment dropout method [9]. The experimental results on the REVERIE [1] and the SOON [2] datasets demonstrate that our proposed method can significantly improve the performance in the GVLN task. In summary, the contributions of our work are as follows:

- We explore a RES-StS method for generating high-quality pseudo labels on sampled trajectories with goal objects to promote the generalization for the GVLN task.
- We propose an encoder-decoder referring expression speaker (RES) model to fully utilize the sequential visual navigation information and predict descriptive instructions. The self-training with a trajectory-instruction match scorer (TIMS) approach is introduced to enhance the performance of RES.
- We present a multi-stage training strategy to integrate the training process of the RES, TIMS, and navigating agent (NA). The experimental results demonstrate that our method achieves new state-of-the-art results on both REVERIE and SOON datasets.

II. Related Work

A. Data Augmentation in VLN and GVLN

In contrast to the Vision-Language Navigation (VLN) [3, 18–23] that asks for the navigation based on the fine-grained instructions, the recent Goal-oriented Vision-Language Navigation (GVLN) task [1, 2] focuses on enabling an agent to search for an instance under the guidance of the target sketch and is considered to be more practical in real life. Some works [2, 4, 6, 7] propose to leverage the memory clue by representing the topological maps or using recurrent units to support global route planning. SIA [24] solves the *where* and *what* problem through understanding high-level instructions. However, the VLN and GVLN tasks' dataset sizes are both small as a result of the simulation environment's specialization, which causes a severe overfitting problem.

In the VLN task, some attempts have been undertaken for data augmentation. For example, the speaker [8, 25, 26] is developed as an inverse model to construct pseudo instructions in accordance with the trajectory presented. Some methods are suggested to imitate more environments through random dropout of features [9], GAN-generated images [27], and random environmental mixup [28]. CITL [29] enhances the trajectories and instructions to improve the model's capacity for representation. AirBERT [22] and VLN-BERT [30] improve the performance by collecting a large number of image-text pairs from the website. Among the above methods, the speaker framework has been most frequently used due to its plug-and-play features. Overall, it is convenient to construct a speaker model to extend the dataset with in-domain pseudo labels. However, all of the existing VLN speaker models are unable to effectively handle goal-oriented instructions. First, the speaker models in VLN only take into account the image features as inputs and completely disregard the object-level information, which is indeed the distinction between the GVLN and VLN. Additionally, most of the previous speaker models adopt the LSTM-based structure to encode and decode instruction words, which makes it difficult to understand the long-term dependence of the entire vision-based trajectory. Therefore, we propose a stronger referring expression speaker (RES) baseline that adapts to the GVLN task. Different from the existing speaker in VLN that can only generate instructions based on step-by-step actions, our RES can fully fuse the sequential environment and object features and describe the goal object in detail. Additionally, we suggest a multi-stage training procedure in which we first train the speaker under a self-training manner with the supervision of the proposed TIMS model, and then apply the speaker in different stages to provide static and dynamic data augmentation for NA.

B. *Self-training*

Recently, self-training has shown its power in main research domains, such as image feature extraction [31, 32], image content understanding [11, 33, 34], and natural language processing [16, 35]. The core of the self-training is to first train a teacher model based on the small labeled dataset, and then use the teacher model to generate pseudo labels on a large number of the unlabeled dataset so that it can be used to retrain a student model. Zoph *et al.* [15] investigate that self-training can improve performance upon pre-training in some practical cases. Furthermore, considering the generation noise by the predicted pseudo labels might mislead the training process and hurt the learning behavior [36], some methods attempt to ensure the reliability of the pseudo labels via the adversarial framework [37] and the action curriculum pseudo labeling [38]. Some methods also use label smoothing [39] or confidence regularization [40] to improve the generalization of the self-training. In this work, we propose to improve the generalization of the RES through the self-training strategy by utilizing the easy-obtained sampled trajectories. In addition, we also design an individual scoring network based on the contrastive loss to filter the imprecise pseudo labels.

C. *Contrastive Learning*

Contrastive Learning [41–43] has shown promising results on cross-modal learning. The basic purpose of contrastive loss is to promote representations to be close for similar samples and distant for dissimilar ones. Some recent works design the network based on contrastive learning to explore the alignment between two types of features [44–46]. Specifically, Huang *et al.* [41] propose a LSTM-based discriminator and some artificial altering strategies to mine negative paths for analyzing the effective proportion of augmented data. Zhao *et al.* [44] discover the typical metrics are insufficient to evaluate grounded path instructions and use the LSTM structure to build a compatibility model based on the contrastive loss and classification loss. Inspired by their works, we build a novel TIMS model in GVLN to filter the low-quality pseudo-data pairs by calculating the trajectory-instruction similarity score. Following the operations of the recent popular CLIP model [17], we take the transformer encoder [10] structure as the backbone and construct positive and negative pairs in the batch. To equip the TIMS with the capacity to recognize the target object, we employ the proposed EOF-module to encode the visual representations. An attention mechanism is also implemented to effectively integrate the dual features into acceptable dimensions so that we can project them to the space where the symmetric cross-entropy loss is applied.

III. Problem Background

Simulator and Environment Settings. Based on the Matterport3D simulator [47], the environment is denoted as a connected graph $G = \{P, \xi\}$, where P and ξ represent navigable nodes and connectivity edges, separately. Formally, the simulator provides the navigable nodes and their panorama composed of 36 images $V = \{v_i\}_{i=1}^{36}$. There are two types of information to be used to represent the visual observations: environment features E and object features O. At each time, the agent can choose one of the adjacent candidates from the current node and navigate to the chosen direction. Then the agent will update its observed surrounding environment features.

Navigation Agent (NA). The GVLN navigation agent aims to autonomously explore an unstructured environment, relying only on vision, and navigate to the location of the target item indicated by the instructions and point out the location of the target object. Specifically, the goal can be divided into two sub-tasks: navigation and localization. The navigation is considered to be completed if the last position a_T in the predicted action sequences $A = \{a_1, a_2, ..., a_T\}$ close to the target (<3m). The localization is considered to be successful if the agent locates [2] or chooses [1] the goal object correctly. The probability of NA prediction can be formulated as $P_A(a_1, ..., a_N | I, E, O) = \prod_{i=1}^{N} P_A(a_i | a_1, ..., a_{i-1}, I, E_i, O_i)$, where a_i, I, E_i, O_i denote the i-th action prediction, instruction, environment features, and object features, respectively.

To solve this challenging task, the base agent leverages cross-modality attention modules for aligning instructions, visual environments, and objects. The encoder-decoder structure is used to model context history and predict actions in each step. The projection head is adopted to point out the target object at the end of the navigation. In this paper, we choose two recent models HAMT [6] and DUET [7] as our NA frameworks. HAMT encodes the trajectory using a hierarchical vision transformer and considers the relationships in time across historical panoramas. DUET suggests a dual-scale graph transformer for action planning and cross-modal understanding.

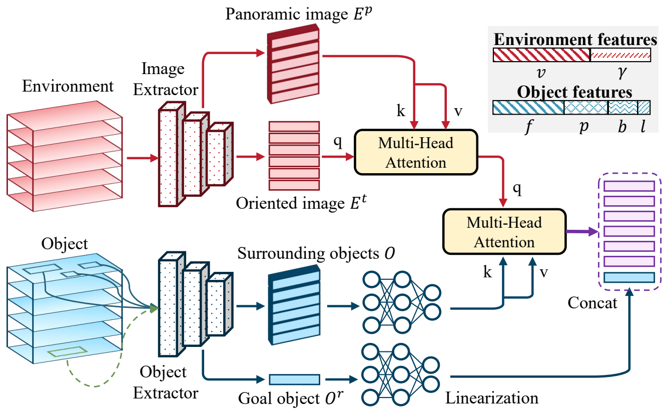


Fig. 2. Illustration of EOF. The fusion of environment features and object features is proposed to equip the model with a stronger ability to perceive and discern the navigation trajectory and the target goal.

Speaker. The speaker serves as an inverse model to the NA, which predicts a set of words $I = \{w_1, w_2, ..., w_L\}$ where w_l means the l-th word in the sentence of length L, to describe the trajectory and annotate sampled data with pseudo-labels. Therefore, the inputs and outputs of the speaker are opposite to those of NA. How to build such a model to effectively encode the long-horizon environment and object representations for the GVLN task remains an open problem. The probability of word prediction can be written as $P_S(w_1, ..., w_L|A, E, O) = \prod_{j=1}^{L} P_S(w_j|w_1, ..., w_{j-1}, A, E, O)$, where w_j denotes the j-th word in the instruction.

IV. Referring Expression Speaker

In this section, we construct RES to dynamically provide goal-oriented instructions using the wealth of visual trajectory cues. The overview architecture of RES is illustrated in Fig. 3. In the spatial domain, we design an EOF module to incorporate panoramic and oriented environment and object features so that the model can leverage global and local information simultaneously. In the temporal domain, we adopt the encoder-decoder transformer as the core of our network to capture sequential features in the long-term distance.

A. Environment-and-Object Fusion (EOF)

As shown in Fig. 2, the EOF module is proposed to facilitate the model to utilize decisive information about nearby environments from different prospects. The difficulty lies in figuring out how to adequately combine the various inputs and eliminate the extraneous elements because the visual components are largely redundant.

First, to supplement the dominating oriented image $E^t \in \mathbb{R}^{N\times d_e}$ and equip the model with the ability to understand the global and local environment features, we explore employing the panoramic images $E^p \in \mathbb{R}^{N\times 36\times d_e}$. The environment features $E = \{V;\gamma\}$ include visual features V and offset angles γ to depict the spatial relationships between each patch image. Specifically, the offset orientation follows $\gamma = (\sin\theta, \cos\theta, \sin\phi, \cos\phi)$, where θ and ϕ denote the heading and elevation direction, respectively. By designating E^t as the query and E^p as the key and value, we adopt the multi-head attention mechanism (MHA) from [10], which is formulated as Equation (1)-(3), to fuse the global environments based on local-oriented observation:

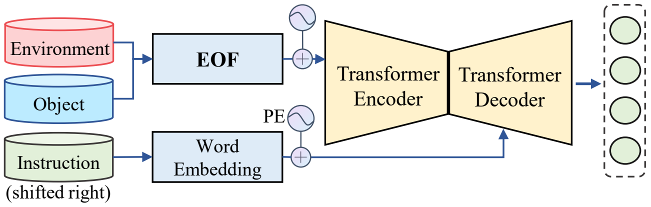


Fig. 3. Illustration of RES. Combined with EOF and transformer architecture, it shows a strong ability for goal-oriented instruction generation. PE denotes the sine and cosine position embedding function.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

$$\text{head}_i = \text{Attention}(Q_m W_i^q, K_m W_i^k, V_m W_i^v) \tag{2}$$

$$\text{MHA}(Q_m, K_m, V_m) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W_H \tag{3}$$

where $Q_m \in \mathbb{R}^{L_q\times d_Q}$, $K_m \in \mathbb{R}^{L_k\times d_K}$, and $V_m \in \mathbb{R}^{L_v\times d_V}$ are queries, keys and values, respectively. L_q, L_k and L_v are sequence lengths of inputs. Let d_k and h denote the intermediate dimension and the number of heads, respectively. The projections are parameters matrices $W_i^q \in \mathbb{R}^{d_Q\times d_k}$, $W_i^k \in \mathbb{R}^{d_K\times d_k}$, $W_i^v \in \mathbb{R}^{d_V\times d_k}$ and $W_H \in \mathbb{R}^{hd_k\times d_m}$, where d_m means the dimension of model.

The diversity of features is further enriched by adding object features $O \in \mathbb{R}^{N\times M\times d_o}$, where M denotes the number of objects of each viewpoint so that the model can use a more thorough visual expression. The regular object visual features f, object position p, and bounding box b are adopted to represent the object. We additionally combine the label l with the aforementioned three categories of features since the label holds the semantic information of the item, which is crucial for comprehending the goal object. Concretely, we provide the categories numerical indices and encode them with an embedding layer,

$$O = \Phi_f(f) + \Phi_p(p) + \Phi_b(b) + \Phi_l(Idx(l)) \tag{4}$$

where Φ means the linear transformation and Idx denotes the quantification process of labels. The object features are then blended with the fused environment features using a new multi-head attention module. The raised fusion step realizes the effective aggregation of the input trajectory features. The target object's information at the destination is essential and shouldn't be disrupted by other environmental circumstances on the way since it directly contributes to the purpose of referring to expression. To this end, we fuse the trajectory features in parallel with the target object features $O^r \in \mathbb{R}^{d_o}$. The full fusion process of EOF is formulated as follows:

$$S_1 = \text{MHA}(E^p W^p, E^t W^t, E^t W^t) \tag{5}$$

$$S_2 = \text{MHA}(S_1 W^s, OW^o, OW^o) \tag{6}$$

$$\widetilde{O}^r = O^r W^r + b^r \tag{7}$$

$$S_0^f = \text{Concat}(S_2, \widetilde{O}^r) \tag{8}$$

where W and b are learnable parameters to project the input features to network hidden space. $S_0^f \in \mathbb{R}^{(N+1)\times d_h}$ is the output feature of the EOF.

B. Encoder-Decoder Transformer Network

1) Vision Encoder: Since a trajectory often includes multiple navigation points, the sine and cosine functions PE_{pos} [10] are used to add the position encoding to the input embedding S_f. Then it is fed to the stacked encoder module composed of identical layers. The MHA and the position-wise feed-forward network (FFN) are the two sub-layers that make up each encoder layer. Specifically, FFN includes two fully-connected layers with the nonlinear activation in between as follows:

$$\text{FFN}(x) = \delta(xW_1^f + b_1^f)W_2^f + b_2^f \tag{9}$$

where δ denotes the ReLU non-linearity and W and b are learnable parameters. The input and output dimensions are the same d_h, and the dimension of the middle linear layer is twice the input dimension. The residual connection [48] and layer normalization [49] are adopted to increase learning stability. This design encodes representations about all previous time steps in addition to the information about the present time step. The procedure of each encoder layer is expressed as:

$$\widetilde{S}_l^f = \text{LayerNorm}(S_{l-1}^f + \text{MHA}(S_{l-1}^f, S_{l-1}^f, S_{l-1}^f)) \tag{10}$$

$$S_l^f = \text{LayerNorm}(\text{FFN}(\widetilde{S}_l^f) + \widetilde{S}_l^f) \tag{11}$$

2) Linguistic Decoder: The structure of the decoder is basically the same as the encoder, except for the addition of a shifting mask to the supervised sentences to ensure each predicted word depends only on the previous ones, and the insertion of another sub-layer that performs attention over the output of the encoder. Supposed I_0 denotes the word embeddings and S_t^f means the output of the vision encoder, the procedure of each decoder layer can be formulated as:

$$\widetilde{I}_l^f = \text{LayerNorm}(I_{l-1} + \text{MHA}(I_{l-1}, I_{l-1}, I_{l-1})) \tag{12}$$

$$\hat{I}_l^f = \text{LayerNorm}(\widetilde{I}_l^f + \text{MHA}(\widetilde{I}_l^f, S_t^f, S_t^f)) \tag{13}$$

$$I_l = \text{LayerNorm}(\text{FFN}(\hat{I}_l^f) + \hat{I}_l^f) \tag{14}$$

After several stacked decoder layers, the output features I_t are projected to a d_w-dimensional space, where d_w is the vocabulary size. The softmax function is used to predict the probability of the next word:

$$p(w_t) = softmax(I_t W^w + b^w) \tag{15}$$

where $W^w \in \mathbb{R}^{d_m \times d_w}$ and $d \in \mathbb{R}^{d_w}$ are learnable parameters.

3) Supervised Loss: Let θ and w^* denote the network parameters and the ground-truth word, respectively. The cross-entropy function used as the objective for the RES is formulated as Equation (16):

$$\mathcal{L}^t = -\sum_{i=1}^{L} \log(f_\theta(w_i^* | w_{1:i-1}^*, E^p, E^t, O, O^r)) \tag{16}$$

V. Self-training Approach

Although RES is intended to enrich data pairs and alleviate the overfitting problem for the NA caused by the small dataset, the training of RES itself may experience the same issue. Therefore, we propose to use the self-training method to eliminate the limitation of small data sets on RES. To the best of our knowledge, we are the first to investigate self-training for the speaker. Furthermore, for improving the quality of pseudo labels, we propose TIMS, which is detailed in Sec. V-A, to determine how similar trajectories and projected instructions are in order to eliminate the low-quality pseudo labels. The overall algorithm of self-training with TIMS for RES is given in Sec. V-B.

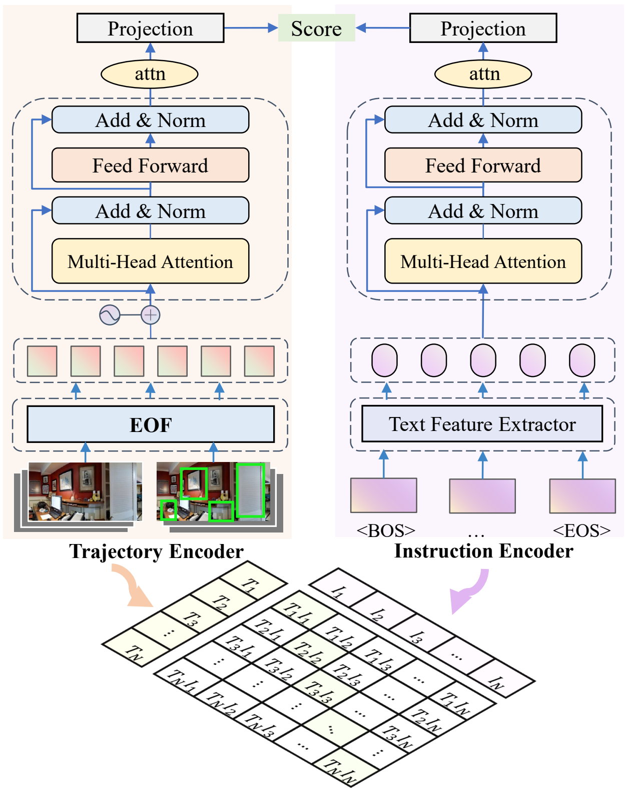


Fig. 4. Illustration of TIMS. A dual-encoder structure is used to individually encode the trajectory and instruction features and determine their similarity score at the end through projection heads.

A. Trajectory-Instruction Matching Scorer (TIMS)

In this section, we propose a dual-encoder structure model to encode the trajectory T and its associated natural language instruction I separately. The cosine similarity score of the two modalities is determined by a single dot product in the learned joint embedding space, and they are only permitted to interact at the top of the network. The illustration of TIMS is shown in Fig. 4. Motivated by the setup in CLIP [17], we train our TIMS to predict which of the $B \times B$ possible (T,I) pairings across a batch actually occurred, where B denotes the batch size. Thus, it is convenient to obtain both B positive pairs and $B^2 - B$ negative pairs.

1) Instruction Encoder: A lightweight and fast DistilBERT model [50] is first applied to encode the words in the instruction $I = \{w_l\}_{l=1}^{L}$. Then the key problem lies in how to effectively represent the high-level semantic content of instruction embeddings. We propose to use a transformer encoder module to strengthen sentence representation and adapt the text embeddings to the dataset at hand. This differs from the previous methods that directly take the first *[CLS]* token in [4] or apply BLSTM to the last hidden layer in [41, 44].

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2022.3233554

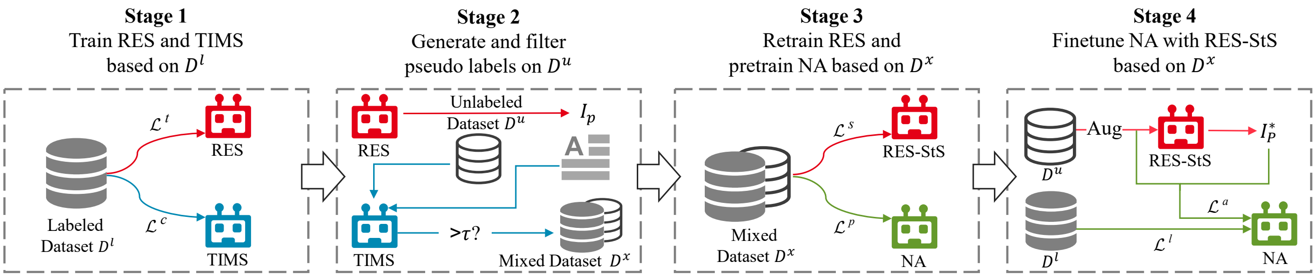


Fig. 5. Overview of the proposed multi-stage training strategy composed of four stages in total. In stage 1, we train the RES and TIMS on the labeled dataset. In stage 2, we use the teacher RES to generate pseudo labels on the sampled unlabeled dataset and filter the low-quality predictions through the trained TIMS. Then the mixed dataset is used to retrain the student RES and train the NA following the pre-training procedure. Finally, we use the trained RES-StS to fine-tune the NA based on the back-translation method.

We discover that since the trained BERT already includes the position encoding of the inputs, it is unnecessary to add another position encoding to the hidden representations. Let $H \in \mathbb{R}^{L\times 768}$ denote the output predicted by the transformer encoder layers. Additionally, we further introduce an attention mechanism [51] to weighted sum the embeddings since self-attention is unable to lower the length of the sequence:

$$M = \tanh(H) \tag{17}$$

$$\alpha = softmax(MW^m) \tag{18}$$

$$h^* = \tanh(\alpha^T H) \tag{19}$$

where $W^m \in \mathbb{R}^{d_h\times 1}$ is a trained parameter and $h^* \in \mathbb{R}^{1\times d_h}$ is an aggregated representation of natural language instructions.

2) Trajectory Encoder: The trajectory encoder in TIMS is built in a manner reminiscent of the RES encoder construction described in Sec IV-A. We employ EOF specifically to fully merge the object features with global and local environmental features. Since EOF mainly focuses on the spatial domain of the visual inputs, we further adopt a transformer encoder module to strengthen the connections within the temporal domain. Additionally, an attention mechanism in Equation (17)-(19) is applied to abbreviate the length of trajectory features because one path typically involves several navigable nodes. This can make it easier to achieve the representations of the two modalities through the dot product.

3) Projection Head: To map the features to the comparison space, a projection head is defined in Equation (20):

$$G(x) = LayerNorm(X + W_2^g(\sigma(W_1^g X))) \tag{20}$$

where W_1^g and W_2^g are learnable parameters, and σ denotes the GELU non-linearity. X is used to present features output by one of the encoders. The residual connection and the layer-norm function are adopted to stable the network's forward and backward propagation. The projection heads are independently applied at the top of the two branches. We use I^e and T^e to denote the final representations of the instruction and the trajectory, respectively.

4) Contrastive Loss: The objective of TIMS is to maximize the similarity scores between positive pairs, which can be constructed as:

$$\begin{aligned}\mathcal{L}^c = &-\frac{1}{2B}\sum_{j=1}^{B}\log\frac{\exp(\langle I_j^e, T_j^e\rangle/t)}{\sum_{k=1}^{B}\exp(\langle I_j^e, T_k^e\rangle/t)}\\ &-\frac{1}{2B}\sum_{k=1}^{B}\log\frac{\exp(\langle I_k^e, T_k^e\rangle/t)}{\sum_{j=1}^{B}\exp(\langle I_j^e, T_k^e\rangle/t)}\end{aligned} \tag{21}$$

where B and $\langle\cdot,\cdot\rangle$ denote the batch size and the inner product, respectively. t represents the temperature parameter [52].

B. Self-training with TIMS

Self-training is a practical approach to incorporating unlabeled data into supervised learning, which has three main steps [14, 16]: 1) train a teacher model on the small labeled dataset, 2) generate pseudo labels using the teacher model on a large amount of unlabeled dataset, 3) train a student model on the mixed dataset. Inspired by this, we propose to generate pseudo-labels for training NA models while also using these pseudo-data pairs to further enhance RES based on self-training. Additionally, the proposed TIMS is used to select the appropriate pseudo labels generated by the rough teacher model f_t. Assumed the output similarity of the TIMS is $f_c(\langle I, T\rangle)$, the loss function of the student model f_s can be expressed as:

$$\mathcal{L}_i^s = -\Psi(f_s(\widetilde{T}_i), f_t(T_i))\mathbb{I}_\tau(f_c(\langle f_t(T_i), T_i\rangle)) \tag{22}$$

where $\tau = \frac{1}{M}\sum_{j=1}^{M} f_c(\langle I_j, T_j\rangle)$ represents the data-driven threshold calculated by M positive labeled samples, Ψ denotes the cross entropy function, and $\widetilde{T}_i$ means the i-th trajectory features with augmentation. The function $\mathbb{I}_\tau$ equals 1 if the similarity of the trajectory-instruction pair is greater than τ, otherwise equals 0. Overall, the completed loss for RES is formulated in Equation (23):

$$\mathcal{L}^{RES} = \mathcal{L}^t + \mathcal{L}^s \tag{23}$$

VI. Multi-stage Training Strategy

Our method consists of three networks in total: a speaker, a scorer, and a navigation agent. As shown in Fig. 5, we propose a multi-stage training strategy to synchronize the entire training process. This is different from previous VLN methods [4, 53] using speaker only in their pre-training stage since their speaker keeps consistent all the time. As we introduce the self-training strategy for RES, the teacher and student RES inherently can generate different pseudo labels during different stages, which can further boost the generalization ability of NA. Let $\mathcal{D}^l$ and $\mathcal{D}^u$ denote labeled and unlabeled dataset, respectively. The process is as follows:

- Stage 1: Train teacher RES and TIMS on $\mathcal{D}^l$, separately. Specifically, RES seeks to learn how to generate descriptive text based on a given trajectory and target, and TIMS learns how to compute the similarity scores of trajectory-text pairings and provide greater matching values to the positive ones.
- Stage 2: Use RES to generate pseudo instructions I_p on $\mathcal{D}^u$ that includes a large number of sampled trajectories without annotations. While the standard metrics of natural language generation cannot be applied without ground truth labels, our TIMS can dynamically calculate similarity scores between data pairs and pick out high-quality pseudo labels. The two datasets are then combined as $\mathcal{D}^x = \mathcal{D}^l \bigcup \mathcal{D}^u$.
- Stage 3: Apply $\mathcal{D}^x$ to retrain RES and pre-train NA. For a fair comparison, we completed the pre-training of NA with the same auxiliary task losses as previous methods did. HAMT adopts the masked language model (MLM), masked region classification (MRC), single-step action prediction (SAP), instruction trajectory matching (ITM), and Spatial Relationship Prediction (SPREL) as auxiliary tasks, while DUET uses the first three tasks with another object grounding (OG) for pre-training. For simplicity, we refer to the pretraining loss of NA as $\mathcal{L}^p$.
- Stage 4: Fine-tune NA to achieve the specific goal of the task at hand. RES-StS is used to further expand the available in-domain data pairs based on the environment dropout approach [9]. By sharing dropout locations across RES and NA, it is able to enrich the variety of environments and instructions that NA can contact during fine-tuning. Similarly, we use the same loss type as previous NA models except for the utilization of additional pseudo data pairs. The main distinction between applying RES during pre-training and fine-tuning is that the former can be viewed as a static augmentation process when using the mixed dataset, whereas the latter can be viewed as a dynamic augmentation procedure since the random environment dropout is developed during each iteration. This strengthens the NA model's robustness and generalization by enabling it to interact with various augmented data in the two stages. The pseudocode of the fine-tuning process of NA with the RES is present in Algorithm 1.

Algorithm 1 The fine-tuning process of NA with RES

```
Input: Labeled dataset $\mathcal{D}^l = \{(I^l_i, A^l_i)\}_{i=1}^N$, unlabeled dataset
    $\mathcal{D}^u = \{A^u_j\}_{j=1}^M$, trained RES model $f_s$.
Output: Navigation agent $f_a$.
while not reach the maximum iteration do
    Sample a batch of items from $D^l$ and $D^u$;
    for $I^l \in \mathcal{D}^l_b$ do
        $\hat{A}^l = f_a(I^l)$;
    end for
    for $A^u \in \mathcal{D}^u_b$ do
        $\widetilde{A}^u = \text{Augment}(A^u)$;
        $\hat{I}^u = f_s(\widetilde{A}^u)$;
        $\hat{A}^u = f_a(\hat{I}^u)$;
    end for
    Update $f_a$ to minimize $\mathcal{L}^l$ of $\{(\hat{A}^l, A^l)\}$
    and $\mathcal{L}^a$ of $\{(\hat{A}^u, \widetilde{A}^u)\}$;
end while
return $f_a$.
```

VII. Experiments

A. Experimental Setup

1) Dataset: We evaluate our methods on REVERIE [1] and SOON [2] datasets for GVLN. REVERIE contains 10,466 instructions over 2,353 objects in the training split. The training split of SOON contains 3,085 sets of instructions with 28,015 trajectories over 38 houses. We use the sampled navigation paths provided by PREVALENT [53] as unlabeled data, and randomly select one object at the endpoint as the target goal. In total, we can generate about 665,206 additional instruction-trajectory pairs. After using TIMS to filter the low-quality generation, the number of pseudo labels is about 241,693. We follow the setup in [7] that uses an object detector [54] to obtain candidate object boxes and convert object grounding settings in SOON similar to the settings in REVERIE.

2) Implementation details: For RES, We use 3 transformer encoder layers and 3 decoder layers with 4 heads, and the dimension of each head is 64. The dimension of hidden layers is 512. The batch size is 64. For TIMS, the numbers of transformer encoder layers for two encoders are both 1 with 4 attention heads. The dimensions of the hidden layers and the batch size are 256 and 8, respectively. The temperature parameter is set to 1 during training. The Adam is used to optimize with a learning rate of 5e-5 for 80,000 iterations, both for training RES and TIMS. The weights of student RES are initialized based on the teacher model. We keep the learning rate and the total number of iterations the same for training the teacher and student RES. For a fair comparison, the training details and the features of the images and objects remain the same as in HAMT and DUET. On a single Tesla V100, the time to reach saturation for RES, TIMS, RES-StS and generate augmented data is around 3 h, 9 h, 11 h, and 8 h, respectively. During fine-tuning, the average time of generating pseudo labels with the dynamic environment dropout is about 0.23 s for each batch of 8. Overall, since the configuration of NA training keeps unchanged, our method will train for around 1.6 days longer than original methods.

B. Evaluation Metrics

1) Instruction Generation Metrics: we adopt standard metrics of natural language generation tasks to evaluate the performance of instruction generation, including BLEU [57], ROUGE [58], CIDEr [59] and SPICE [60]. Additionally, a new metric **Hit Rate** (HR) is proposed to determine how well a speaker can capture the object category and room

TABLE I
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE REVERIE DATASET.

Methods	Val Seen					Val Unseen					Test Unseen				
	Navigation			Grounding		Navigation			Grounding		Navigation			Grounding	
	OSR	SR	SPL	RGS	RGSPL	OSR	SR	SPL	RGS	RGSPL	OSR	SR	SPL	RGS	RGSPL
Seq2Seq [3]	35.70	29.59	24.01	18.97	14.96	8.07	4.20	2.84	2.16	1.63	6.88	3.99	3.09	2.00	1.58
RCM [55]	29.44	23.33	21.82	16.23	15.36	14.23	9.29	6.97	4.89	3.89	11.68	7.84	6.67	3.67	3.14
SMNA [56]	43.29	41.25	39.61	30.07	28.98	11.28	8.15	6.44	4.54	3.61	8.39	5.80	4.53	3.10	2.39
FAST-MATTN [1]	55.17	50.53	45.50	31.97	29.66	28.20	14.40	7.19	7.84	4.67	30.63	19.88	11.61	11.28	6.08
SIA [24]	65.85	61.91	57.08	45.96	42.65	44.67	31.53	16.28	22.41	11.56	44.56	30.80	14.85	19.02	9.20
RecBERT [4]	53.90	41.79	47.96	38.23	35.61	35.02	30.67	24.90	18.77	15.27	32.91	29.61	23.99	16.50	13.51
Airbert [22]	48.98	47.01	42.34	32.75	30.01	34.51	27.89	21.88	18.23	14.18	34.20	30.28	23.61	16.83	13.28
HOP [5]	54.88	47.19	13.80	38.65	33.85	36.24	26.11	16.46	18.85	15.73	33.06	24.34	16.38	17.69	14.34
HAMT [6]	47.65	43.29	40.19	27.20	25.18	36.84	32.95	30.20	18.92	17.28	33.41	30.40	26.67	14.88	13.08
HAMT w/ RES-StS (Ours)	**60.65 (+13.00)**	**58.54 (+15.25)**	**56.01 (+15.82)**	**42.66 (+15.46)**	**40.70 (+15.52)**	**37.09 (+0.25)**	**34.25 (+1.90)**	**30.31 (+0.11)**	**20.48 (+1.56)**	**18.09 (+0.81)**	**39.48 (+6.02)**	**37.38 (+6.98)**	**32.65 (+5.98)**	**20.07 (+5.19)**	**17.50 (+4.42)**
DUET [7]	73.86	71.75	63.94	57.41	51.14	51.07	46.98	33.73	32.15	23.03	56.91	52.51	36.06	31.88	22.06
DUET w/ RES-StS (Ours)	**78.50 (+4.64)**	**75.40 (+3.65)**	**67.13 (+3.19)**	**62.08 (+4.67)**	**55.39 (+4.25)**	**55.01 (+3.94)**	**48.85 (+1.87)**	33.07 (-0.66)	**33.17 (+1.02)**	22.33 (-0.70)	**62.41 (+5.5)**	**57.23 (+4.72)**	**38.61 (+2.55)**	**35.33 (+3.45)**	**23.64 (+1.58)**

type compared to the ground truth annotation. The HR is formulated in Equation (24),

$$\text{HitRate}(\hat{y}_i, y_i) = \lambda \sum_{j}^{N} \frac{\text{LLCS}(M_l(\hat{y}_i), M_l(y_{ij})))}{\text{L}(M_l(y_{ij}))} + (1-\lambda) \sum_{j}^{N} \frac{\text{LLCS}(M_t(\hat{y}_i), M_t(y_{ij})))}{\text{L}(M_t(y_{ij}))} \tag{24}$$

where $LLCS$ denotes the length of the longest common subset between two sets and L means the length of the phrase. M_l and M_t represent the expressions of the room type and the goal object, respectively. Here we use the NLTK toolkit [61] to extract the specific contents from sentences. $\hat{y}_i$ and y_{ij} mean the prediction results output by the speaker and the j-th ground truth annotation of the i-th trajectory, respectively, since each path is usually associated with several labeled instructions. λ is used to balance the proportion of two types of expression and the results reported in this paper are all for the case where $\lambda = 0.5$. Additionally, we also use the **Object Hit Rate** (OHR) metric to count the proportion of the correct mentioned objects on the unlabeled dataset, which is formulated as OHR $= \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(M_t(\hat{y}_i), o_i)$, where o_i means the label of the assigned object for the i-th sampled path and $\mathbb{I}$ equals 1 if two phrases are consistent, otherwise equals 0.

2) Scorer Metrics: We define a **Matching Success Rate** (MSR) metric to evaluate the performance of the scorer during training. The formulation of MSR is shown in Equation (25), where B and N denote the batch size and the number of mini-batches in a set, respectively. $\mathbb{I}_{pos}$ equals 1 if the prediction is correct, otherwise equals 0. The MSR can reflect the average proportion of the correct matched pairs predicted by the scorer to the total positive data pairs in every mini-batch.

$$\text{MSR} = \frac{1}{NB} \sum_{i=1}^{N} \sum_{j=1}^{B} \mathbb{I}_{pos}(\arg\max_{k}(f_c(\langle T_{ij}, I_{ik} \rangle))) \tag{25}$$

TABLE II
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE SOON DATASET.

Split	Method	OSR	SR	SPL	RGSPL
Val Unseen	GBE [2]	28.54	19.52	13.34	1.16
	DUET [7]	50.91	36.28	22.58	3.75
	DUET w/ RES-StS (Ours)	**52.92 (+2.01)**	**40.18 (+3.90)**	**26.30 (+3.72)**	**5.29 (+1.54)**
Test Unseen	GBE [2]	21.45	12.90	9.23	0.45
	DUET [7]	43.00	33.44	21.42	4.17
	DUET w/ RES-StS (Ours)	**48.07 (+5.07)**	**38.02 (+4.58)**	**25.17 (+3.75)**	**7.31 (+3.14)**

3) Goal-oriented Navigation Metrics: Following [1, 2], we use the standard metrics to measure navigation and object grounding performance, including Trajectory Success Rate (SR), Oracle SR (OSR), SR penalized by Path Length (SPL), Remote Grounding Success (RGS), and RGS penalized by Path Length (RGSPL). All indicators are the higher the better.

C. Compared to state-of-the-art results

1) REVERIE: Table I compares our method with state-of-the-art models on the REVERIE dataset. The results demonstrate that our method is beneficial for improving the performance of navigating agent models and greatly beats the state-of-the-art on all evaluation metrics. For instance, in the test unseen split, SPL and RGSPL are greater than HAMT at 5.98% and 4.42%, and higher than DUET at 2.55% and 1.58%, respectively. The improvement is consistently observed on other metrics as well. Moreover, using RES-StS to expand the training set has also significantly improved the results in the seen environments, achieving improvements in SR, SPL, and RGSPL of 15.25%, 15.82%, and 15.52%, respectively. In addition, we also observe that there is a large improvement gap between the val-unseen split and the test-unseen split. We hypothesize that the stark disparity in data distribution between the two subsets is to blame. Above all, the experimental results

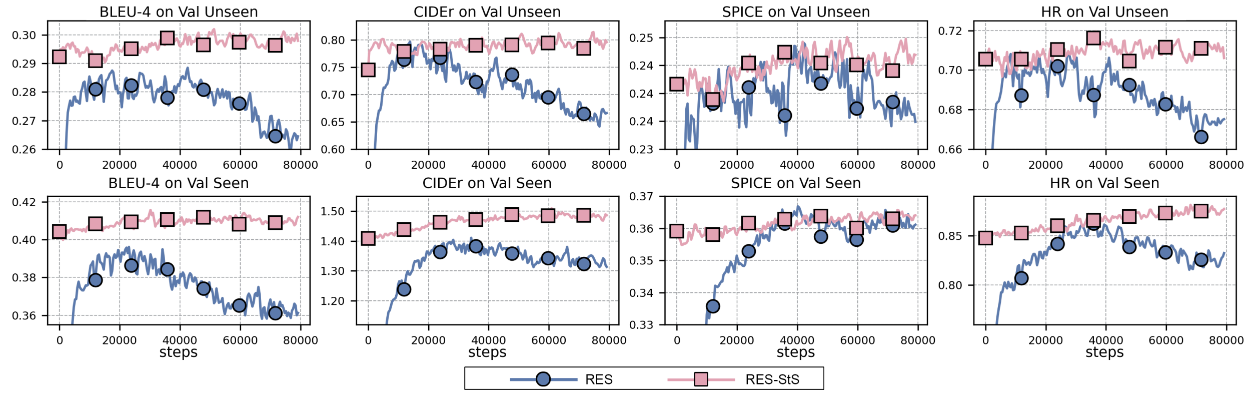


Fig. 6. Learning curves for multiple evaluation metrics for RES on seen and unseen validation splits. It shows that there is a serious overfitting problem when RES is trained on the original REVERIE dataset, resulting in a sharp drop in the metrics on the validation set after a certain number of iterations. When the self-training method is used, this problem is greatly alleviated, and most of the evaluation metrics show a continuous upward trend.

TABLE III
COMPARISON OF DIFFERENT FEATURE COMPONENTS ON THE REVERIE UNSEEN VALIDATION SPLIT. THE "OI", "PI", "SO", AND "GO" STAND FOR ORIENTED IMAGES, PANORAMIC IMAGES, SURROUNDING OBJECTS, AND THE GOAL OBJECT, RESPECTIVELY.

Id	OI	PI	SO	GO	B@1	B@4	ROUGE	CIDEr	SPICE	HR
1	✓				0.526	0.195	0.455	0.339	0.174	0.447
2	✓	✓			0.541	0.219	0.451	0.400	0.190	0.526
3	✓	✓	✓		0.541	0.226	0.458	0.431	0.201	0.511
4	✓		✓	✓	0.585	0.278	0.511	0.739	0.241	0.681
5	✓	✓	✓	✓	**0.596**	**0.294**	**0.515**	**0.766**	**0.244**	**0.705**

TABLE IV
COMPARISON OF SELF-TRAINING WITH AND WITHOUT TIMS FOR RES.

	Method	B@1	B@4	ROUGE	CIDEr	SPICE	HR
Seen	RES	0.692	0.402	0.613	1.435	0.359	0.866
	RES-St	0.698	0.414	0.615	1.490	0.363	0.863
	RES-StS	**0.700**	**0.415**	**0.620**	**1.491**	**0.365**	**0.869**
Unseen	RES	0.596	0.294	0.515	0.766	0.244	0.705
	RES-St	0.598	0.300	0.520	0.803	0.249	0.700
	RES-StS	**0.606**	**0.302**	**0.523**	**0.812**	**0.251**	**0.713**

TABLE V
ABLATION OF LABEL FEATURE FOR RES.

	Method	B@1	B@4	ROUGE	CIDEr	SPICE	HR
Seen	w/o label	0.688	0.395	0.603	1.351	0.349	0.831
	w/ label	**0.692**	**0.402**	**0.613**	**1.435**	**0.359**	**0.866**
Unseen	w/o label	0.594	0.288	0.515	0.744	0.235	0.694
	w/ label	**0.596**	**0.294**	**0.515**	**0.766**	**0.244**	**0.705**

indicate that our approach can unleash the enormous potential of the previous NA models, raising the learning ceiling for the available dataset.

2) *SOON:* Since the number of annotated instructions provided by the SOON dataset is much lower than that provided by REVERIE, and the instructions contain more detailed content, the model tends to encounter greater challenges in learning. Nevertheless, the experimental results shown in Table II prove that using our RES-StS method for data augmentation can effectively improve the model's generalization ability, achieving a new state-of-the-art threshold. In the test unseen split, we successfully improve the previous best model DUET [7] by 5.07% on OSR, 4.58% on SR, 3.75% on SPL, and 3.14% on RGSPL. This reveals that data augmentation is crucial for the data-driven deep learning models, and our RES-StS contributes significantly to the NA models by offering appropriate pseudo labels for the different datasets in the GVLN task.

D. *Quantitative Analysis*

1) *Effect of different feature components:* As described in Sec. IV-A, we use four types of features to enrich the trajectory representations: panoramic images E^p, oriented images E^t, surrounding objects O, and the goal object O^r. EOF is proposed to fuse these complicated features effectively. We explore the importance of the above features contributing to our RES and the result is shown in Table III. It demonstrates that panoramic images and surrounding objects can effectively endow global observations with local representations, improving CIDEr by 0.061 and 0.092, respectively. The goal object information can significantly improve the performance (the CIDEr and HR are increased from 0.431 to 0.766 and from 0.511 to 0.705, respectively) by administering the necessary clues to the model. The reasonable fusion method allows the obtained path features to contain richer linguistic, semantic, and dominant information, providing a solid basis for the construction of RES and TIMS models.

2) *Effect of self-training for RES:* Table IV shows that the self-training approach can improve most metrics in both seen and unseen environments. However, the presence of some noise in the generated pseudo labels affects the model's learning of the key information representation, resulting in some decrease in HR values in both environments (HR 0.866 *vs.* 0.863 on the seen split). This problem was solved when we introduced TIMS to filter low-quality pseudo data pairs before retraining, which validates our conjecture that TIMS can effectively alleviate undesirable noise in the process of generating pseudo labels and avoid its misleading relearning of the student model. As a result, RES-StS outperforms RES

TABLE VI
ABLATIONS OF THE DIFFERENT COMPONENTS OF TIMS.

Id	Method	MSR on Val Seen	MSR on Val Unseen
1	*[CLS]*	90.2	84.4
2	BLSTM	95.0	87.2
3	**Transformer**	**95.7**	**90.4**
4	Mean	95.4	87.6
5	**Attn**	**95.7**	**90.4**
6	PE_s	94.0	90.0
7	PE_l	93.1	86.7
8	**w/o PE**	**95.7**	**90.4**

TABLE VII
ABLATION OF RES AND StS FOR NAVIGATION PERFORMANCE ON THE REVERIE UNSEEN VALIDATION SET.

Stage	RES	StS	OSR	SR	SPL	RGS	RGSPL
Pre-train	×	×	35.44	32.55	22.73	21.76	14.70
	✓	×	37.09	33.88	23.06	21.78	14.91
	✓	✓	**42.23**	**38.71**	**27.21**	**26.16**	**18.29**
Fine-tune	×	×	51.07	46.98	33.73	32.15	23.03
	✓	×	54.70	47.57	32.66	32.89	22.52
	✓	✓	**55.01**	**48.85**	33.07	**33.17**	22.33

in all metrics. To more visually demonstrate the difference between the training effects of the teacher model and the student model, we visualize the learning curves in Fig. 6. It is clear that without self-training, the teacher model has a serious overfitting issue that prevents it from simultaneously reaching the highest checkpoint for the majority of the metrics. After introducing the self-training method with TIMS to retrain a student model, the overfitting is greatly alleviated and most of the indicators are consistently increasing and surpassing the teacher model.

3) Effect of the label on the goal object: As the labels of the objects contain essential semantic information, we compare the effect of adding the label embeddings l into the object features in Table V. It demonstrates that all measures significantly improve with the addition of label features, indicating that direct semantic information may make it simpler for the model to comprehend objects.

4) Effect of the compositions of TIMS: Since the representation of trajectory features has been carefully designed in EOF, we mainly explore how to represent the high-level semantic information of the instruction. From Table VI we can obtain that: (1) The #1 – #3 rows compare the different methods for representing the semantic information of sentences embedded by the trained BERT model. It shows that adding additional encoding modules can capture the semantic features better than directly using a single *[CLS]* token. Specifically, the Transformer encoder can provide a stronger representation ability than the BLSTM does. (2) The #4 – #5 rows compare the different aggregation approaches. The attention mechanism can emphasize the important content more accurately compared to meaning pooling. (3) The #6 – #8 rows explore whether to add another position encoding after using the trained BERT to extract text embeddings. PE_s and PE_l denote the sine and cosine functions [10] and learned positional embeddings [62], respectively. It represents that it is unnecessary to add another position encoding since the trained BERT model has already considered the order of the sequence.

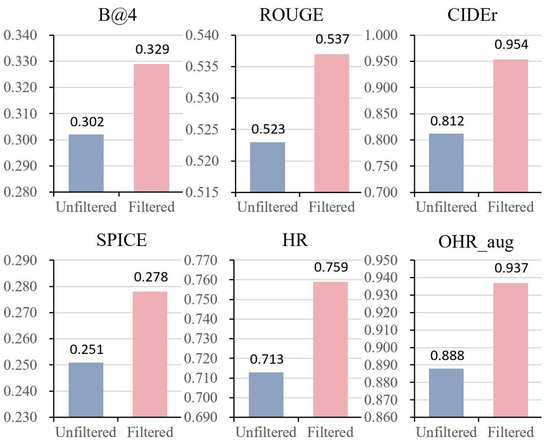
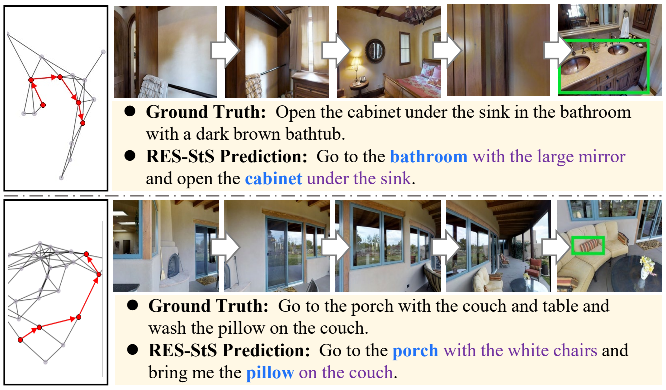


Fig. 7. Illustration of the effect of using TIMS to filter the pseudo labels on the unseen set. OHR_aug means the object hit rate on the augmented data.

5) Effect of using TIMS to filter low-quality predictions: We further verify that whether the TIMS model can meet our requirement to filter out some predictions of low quality based on the trajectories and instructions. As shown in Fig. 7, all metrics on the unseen validation split obtain obvious improvement after applying TIMS to filter the predictions below the threshold. For the essential object and location information, HR increases from 0.713 to 0.759. In addition, we also track the hit rate of objects on our augmented dataset (OHR_aug), which shows an improvement from 0.888 to 0.937. It is evident that our method can capture most of the essential information for GVLN, and TIMS has the capacity to filter out a large number of noisy pseudo labels.

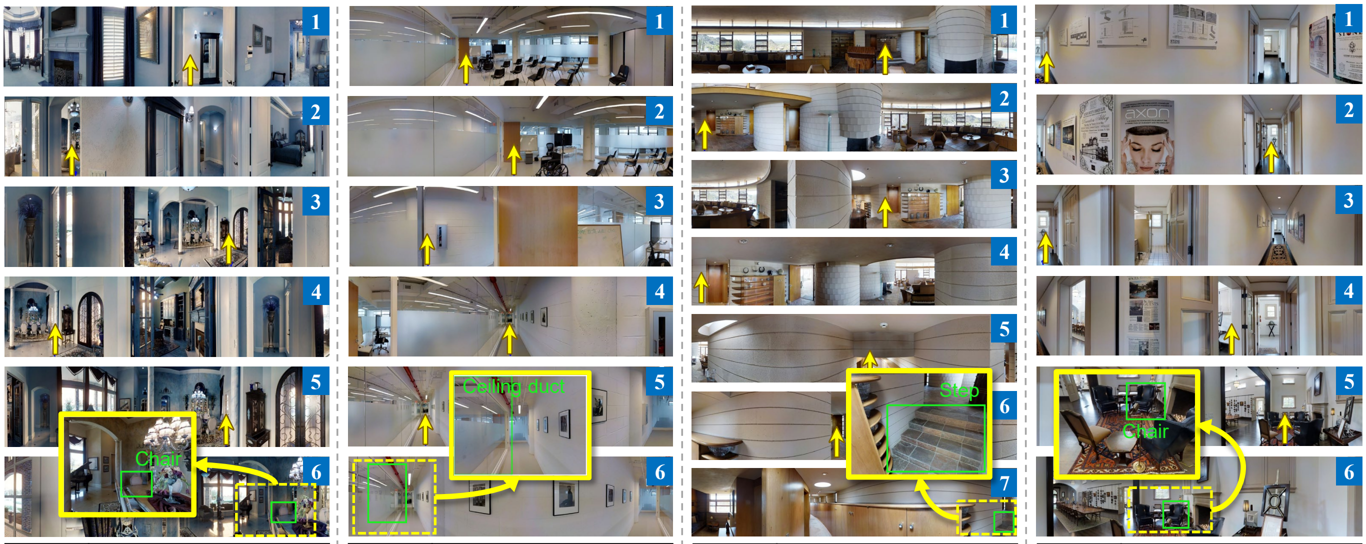


- **Ground Truth:** Open the cabinet under the sink in the bathroom with a dark brown bathtub.
- **RES-StS Prediction:** Go to the **bathroom** with the large mirror and open the **cabinet** under the sink.

- **Ground Truth:** Go to the porch with the couch and table and wash the pillow on the couch.
- **RES-StS Prediction:** Go to the **porch** with the white chairs and bring me the **pillow** on the couch.

Fig. 8. Qualitative examples of predicted instructions compared with the ground truth. It shows that RES-StS can capture the correct key points in the instructions (shown in blue) with appropriate descriptions (shown in purple).

6) Promoting effect of RES-StS on NA: We compare the power of using RES with and without the TIMS-filtering self-training approach to promoting the navigation and localization ability of NA in Table VII. We chose DUET as our experimental subject. Specifically, when both RES and StS are crossed, it indicates the original training method; when RES is ticked and StS is crossed, it represents the data augmentation using RES without self-training; when both RES and StS are



Target	Chair
RES	Go to the dining room with the large round table and pull out the chair closest to the piano.
TIMS	1.2 ($> \tau$)

(a)

Target	Ceiling duct
RES	Go to the hallway with the pictures of a man and clean the TV *(wrong object)*.
TIMS	0.9 ($< \tau$)

(b)

Target	Step
RES	Go to the hallway with the blue *(wrong color)* walls and clean the bottom step.
TIMS	0.9 ($< \tau$)

(c)

Target	Chair
RES	Go to the family room on level 2 and pull out the chair closest to the entrance *(wrong location)*.
TIMS	1.2 ($> \tau$) *(wrong score)*

(d)

Fig. 9. Visualization of some success and failure cases on the sampled trajectories. For each example, the target object, the pseudo labels predicted by RES, and the score (divided by τ) determined by TIMS are represented in the bottom tables. The words marked in purple mean the wrong content with the explanation marked in red. Specifically, (a) presents a successful case that both RES and TIMS predict correctly. (b) and (c) show that RES outputs incorrect contents while TIMS successfully filter them out. (d) is the case that TIMS fails to identify the wrong pseudo instruction.

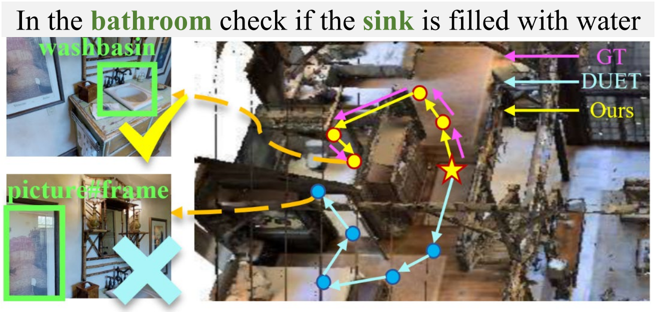


Fig. 10. Comparison of predicted trajectories of ours and the state-of-the-art DUET [7]. The left green boxes show the chosen object at the end of the navigation.

ticked, it means that the adopted RES has been retrained by the TIMS-based self-training method. The results reveal that using RES can effectively improve the robustness of the model in both the pre-training (SR 33.88 *vs.* 32.55) and fine-tuning stages (SR 47.57 *vs.* 46.98). The effect of data augmentation will be further strengthened when we utilize a stronger RES after TIMS-filtering self-training, significantly improving the performance of the original NA model.

E. Qualitive Analysis

1) Referring expression generation: In Fig. 8, we show some visualization examples of the predicted instructions of the trajectories on the REVERIE dataset. Specifically, the

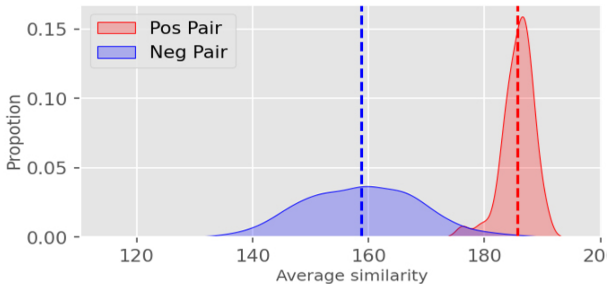


Fig. 11. Illustration of the distribution of the similarity values of positive and negative pairs predicted by TIMS.

description can be generated beyond what is designated in the ground-truth sentences, which shows that our RES-StS has learned how to comprehend visual observations effectively. As for the top trajectory in Fig. 8, the predicted instruction uses the phrase *"with a large mirror"* to describe the bathroom, which is not represented in the ground-truth annotation. However, a large mirror is in fact hang on the wall in the third image, demonstrating the accuracy of our prediction. Therefore, using RES-StS can achieve the goal of expanding the dataset by generating more instructions with different descriptions, which is beneficial to data-driven model learning.

2) Visualization and analysis of some failure cases: In addition to visualizing some successful examples, we also track some failure cases to analyze the limitation of our models in the hope of inspiring future work. In Fig. 9, it shows that although our speaker is capable of describing the objective with visual features, *e.g.*, (a) successfully captures the *large round table* through the path and the *piano* around the *chair*, it may nevertheless forecast incorrect contents in some circumstances. For example, (b) shows that the RES mistakenly outputs the *ceiling duct* to the *TV*, and in (c), it recognizes the *brown* wall as the *blue* one. Fortunately, our TIMS can dynamically identify and remove these errors. The reason for this difference is that the captioning task is much harder than the classification task, and sometimes the grounded object with a similar appearance may confuse the model. However, there are some cases that RES and TIMS are still unable to solve. In (d), the descriptive location of the *chair* is ambiguous since we can see there are other chairs that are closer to the entrance.

Overall, we think the absence of some detailed semantic information and relative spatial coordinates between objects may contribute to the above problems. Some recent image captioning works that consider the relative relationship between objects [63, 64] could possibly overcome this shortcoming and should be further explored in future works.

3) Navigation trajectory: A visualized example of predicted trajectories and the ground truth path is shown in Fig. 10. For the instruction *"In the bathroom check if the sink is filled with water"*, it is demonstrated that our model succeeds in completing the objective where the original DUET model fails. This implies that data augmentation during training enhances the model's combined capacity to navigate and localize by enabling it to view a greater variety of targets and learn how to find them in the scene.

4) Distribution of similarity values of positive and negative pairs: In Fig. 11, we visualize the distribution of similarity values of positive and negative pairs predicted by TIMS. As described in Sec. V-B, we use the average similarity value of positive examples as the threshold τ for filtering low-quality generated pseudo data pairs. Experimental results show that although the total number of available samples is decreased, the self-training for RES is more effective since the noisy generation is eliminated. Statistics show that just 1.3% of negative example pairings have similarity values above this threshold, demonstrating the reliability of using TIMS to choose the generated natural language instructions that are appropriate for the path and the goal object.

VIII. Conclusion

We propose a RES-StS approach for generating high-quality pseudo labels on the sampled unlabeled data to address the challenge caused by the small dataset in the GVLN task. A referring expression speaker (RES) model is designed to predict goal-oriented natural language descriptions. To enable the model to effectively process the sequential visual observations, we present EOF to fuse four types of inputs and adopt the transformer architecture as the core of our encoder-decoder network. We propose to use a self-training strategy for improving the performance of RES so that a large number of unlabeled trajectories that are easy to sample can be fully utilized. Considering the potential disturbance caused by the rough generation instructions, we present TIMS with a dual-encoder structure to filter the pseudo labels to diminish the noise. Finally, we summarize the whole training process for RES, TIMS, and the navigating agent as a multi-stage training strategy. Experimental results verify the effectiveness of our proposed components and demonstrate that our method can significantly unleash the potential of previous methods, achieving state-of-the-art performance on GVLN benchmarks REVERIE and SOON.

We believe that our approach has good expansibility and robustness and can also serve other VLN-like tasks as well. Nevertheless, since the addition of individual networks would require extra training time, further investigation needs to further investigate on how to build such a system in an end-to-end manner with higher efficiency.

References

[1] Y. Qi, Q. Wu, P. Anderson, X. Wang, W. Y. Wang, C. Shen, and A. v. d. Hengel, "Reverie: Remote embodied visual referring expression in real indoor environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9982–9991.

[2] F. Zhu, X. Liang, Y. Zhu, Q. Yu, X. Chang, and X. Liang, "Soon: Scenario oriented object navigation with graph-based exploration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 12 689–12 699.

[3] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3674–3683.

[4] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, "Vln bert: A recurrent vision-and-language bert for navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1643–1653.

[5] Y. Qiao, Y. Qi, Y. Hong, Z. Yu, P. Wang, and Q. Wu, "Hop: History-and-order aware pre-training for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 418–15 427.

[6] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, "History aware multimodal transformer for vision-and-language navigation," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[7] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Think global, act local: Dual-scale graph transformer for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 537–16 547.

[8] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-follower models for vision-and-

language navigation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[9] H. Tan, L. Yu, and M. Bansal, "Learning to navigate unseen environments: Back translation with environmental dropout," *arXiv preprint arXiv:1904.04195*, 2019.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[11] G. Zhou, D. Wang, Y. Yan, H. Chen, and Q. Chen, "Semi-supervised 6d object pose estimation without using real annotations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5163–5174, 2022.

[12] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.

[13] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[14] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 687–10 698.

[15] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, "Rethinking pre-training and self-training," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 3833–3845.

[16] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "St++: Make self-training work better for semi-supervised semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4268–4277.

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[18] W. Zhang, C. Ma, Q. Wu, and X. Yang, "Language-guided navigation via cross-modal grounding and alternate adversarial learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[19] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Wang, and L. Zhang, "Vision-language navigation policy learning and adaptation," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[20] D. An, Y. Qi, Y. Huang, Q. Wu, L. Wang, and T. Tan, "Neighbor-view enhanced model for vision and language navigation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5101–5109.

[21] T. Wang, Z. Wu, and D. Wang, "Visual perception generalization for vision-and-language navigation via meta-learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[22] P.-L. Guhur, M. Tapaswi, S. Chen, I. Laptev, and C. Schmid, "Airbert: In-domain pretraining for vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1634–1643.

[23] B. Lin, Y. Zhu, Z. Chen, X. Liang, J. Liu, and X. Liang, "Adapt: Vision-language navigation with modality-aligned action prompts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 396–15 406.

[24] X. Lin, G. Li, and Y. Yu, "Scene-intuitive agent for remote embodied visual grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7036–7045.

[25] S. Agarwal, D. Parikh, D. Batra, P. Anderson, and S. Lee, "Visual landmark selection for generating grounded and interpretable navigation instructions," in *CVPR workshop on Deep Learning for Semantic Visual Navigation*, 2019.

[26] T.-J. Fu, X. E. Wang, M. F. Peterson, S. T. Grafton, M. P. Eckstein, and W. Y. Wang, "Counterfactual vision-and-language navigation via adversarial path sampler," in *European Conference on Computer Vision*. Springer, 2020, pp. 71–86.

[27] J. Li, H. Tan, and M. Bansal, "Envedit: Environment editing for vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 407–15 417.

[28] C. Liu, F. Zhu, X. Chang, X. Liang, Z. Ge, and Y.-D. Shen, "Vision-language navigation with random environmental mixup," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1644–1654.

[29] X. Liang, F. Zhu, Y. Zhu, B. Lin, B. Wang, and X. Liang, "Contrastive instruction-trajectory learning for vision-language navigation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 1592–1600.

[30] A. Majumdar, A. Shrivastava, S. Lee, P. Anderson, D. Parikh, and D. Batra, "Improving vision-and-language navigation with image-text pairs from the web," in *European Conference on Computer Vision*. Springer, 2020, pp. 259–274.

[31] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 132–149.

[32] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," *arXiv preprint arXiv:1911.05371*, 2019.

[33] L. Yu, X. Liu, and J. van de Weijer, "Self-training for class-incremental semantic segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[34] W. Wang, T. Lin, D. He, F. Li, S. Wen, L. Wang, and J. Liu, "Semi-supervised temporal action proposal generation via exploiting 2-d proposal map," *IEEE Trans-*

actions on Multimedia, vol. 24, pp. 3624–3635, 2022.

[35] S. Zhu, R. Cao, and K. Yu, "Dual learning for semi-supervised natural language understanding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1936–1947, 2020.

[36] L. Yu, X. Liu, and J. van de Weijer, "Self-training for class-incremental semantic segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

[37] C. Cao, T. Lin, D. He, F. Li, H. Yue, J. Yang, and E. Ding, "Adversarial dual-student with differentiable spatial warping for semi-supervised semantic segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[38] A. Tong, C. Tang, and W. Wang, "Semi-supervised action recognition from temporal augmentation using curriculum learning," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[39] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, "Regularizing neural networks by penalizing confident output distributions," *ArXiv*, vol. abs/1701.06548, 2017.

[40] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.

[41] H. Huang, V. Jain, H. Mehta, J. Baldridge, and E. Ie, "Multi-modal discriminative model for vision-and-language navigation," *arXiv preprint arXiv:1905.13358*, 2019.

[42] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[43] L. Tao, X. Wang, and T. Yamasaki, "An improved inter-intra contrastive learning framework on self-supervised video representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5266–5280, 2022.

[44] M. Zhao, P. Anderson, V. Jain, S. Wang, A. Ku, J. Baldridge, and E. Ie, "On the evaluation of vision-and-language navigation instructions," *arXiv preprint arXiv:2101.10504*, 2021.

[45] T. Han, W. Xie, and A. Zisserman, "Temporal alignment networks for long-term video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2906–2916.

[46] J. Cho, S. Yoon, A. Kale, F. Dernoncourt, T. Bui, and M. Bansal, "Fine-grained image captioning with clip reward," in *NAACL-HLT*, 2022.

[47] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv preprint arXiv:1709.06158*, 2017.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *ArXiv*, vol. abs/1607.06450, 2016.

[50] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[51] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, 2016, pp. 207–212.

[52] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2495–2504.

[53] W. Hao, C. Li, X. Li, L. Carin, and J. Gao, "Towards learning a generic agent for vision-and-language navigation via pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 137–13 146.

[54] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[55] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6629–6638.

[56] C.-Y. Ma, J. Lu, Z. Wu, G. AlRegib, Z. Kira, R. Socher, and C. Xiong, "Self-monitoring navigation agent via auxiliary progress estimation," *arXiv preprint arXiv:1901.03035*, 2019.

[57] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[58] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[59] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[60] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *European conference on computer vision*. Springer, 2016, pp. 382–398.

[61] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[62] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning

Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1243–1252.

[63] Z.-J. Zha, D. Liu, H. Zhang, Y. Zhang, and F. Wu, "Context-aware visual policy network for fine-grained image captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 710–722, 2022.

[64] Z. Shao, J. Han, D. Marnerides, and K. Debattista, "Region-object relation-aware dense captioning via transformer," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2022.

Liuyi Wang received the bachelor's degree in control science and engineering from Tongji University, Shanghai, China, in 2020, where she is currently pursuing the Ph.D. degree. Her research interests include deep learning, multi-modal fusion, and vision-and-language navigation.

Zongtao He received the bachelor's degree in control science and engineering from Tongji University, Shanghai, China, in 2019. He is currently pursuing a Ph.D. in control science and engineering at Tongji University, Shanghai, China. His research interests include vision-and-language navigation, deep learning, and multimodal processing.

Ronghao Dang is currently pursuing the M.S. degree with the Department of Control Science and Engineering, College of Electronics and Information Engineering, Tongji University. His research interests include action recognition and Vision-and-Language Navigation.

Huiyi Chen is a researcher and designer working in the intersection of design and technology. She is an adjunct Faculty at Rutgers University. She received B.A degree in psychology in University of Southern California in 2016. She received her M.P.S degree in Interactive Telecommunication Program at New York University in 2019 and she served as a postdoc fellow at the same place from 2019 - 2020. Her research interests include human-computer interfaces and interaction, multimedia for virtual reality and augmented reality.

Chengju Liu received the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2011. From October 2011 to July 2012, she was with the BEACON Center, Michigan State University, East Lansing, MI, USA, as a Research Associate. From March 2011 to June 2013, she was a Postdoctoral Researcher with Tongji University, where she is currently a Professor with the Department of Control Science and Engineering, College of Electronics and Information Engineering, and a Chair Professor of Tongji Artificial Intelligence (Suzhou) Research Institute. She is also a Team Leader with the TJArk Robot Team, Tongji University. Her research interests include intelligent control, motion control of legged robots, and evolutionary computation.

Qijun Chen received the B.S. degree in automation from Huazhong University of Science and Technology, Wuhan, China, in 1987, the M.S. degree in information and control engineering from Xi'an Jiaotong University, Xi'an, China, in 1990, and the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 1999. He is currently a Full Professor in the College of Electronics and Information Engineering, Tongji University, Shanghai, China. His research interests include robotics control, environmental perception, and understanding of mobile robots and bioinspired control.