CrossMap Transformer: A Crossmodal Masked Path Transformer Using Double Back-Translation for Vision-and-Language Navigation

Aly Magassouba[1], Komei Sugiura[2], and Hisashi Kawai[1]

Abstract—Navigation guided by natural language instructions is particularly suitable for Domestic Service Robots that interacts naturally with users. This task involves the prediction of a sequence of actions that leads to a specified destination given a natural language navigation instruction. The task thus requires the understanding of instructions, such as "Walk out of the bathroom and wait on the stairs that are on the right". The Visual and Language Navigation remains challenging, notably because it requires the exploration of the environment and at the accurate following of a path specified by the instructions to model the relationship between language and vision. To address this, we propose the CrossMap Transformer network, which encodes the linguistic and visual features to sequentially generate a path. The CrossMap transformer is tied to a Transformer-based speaker that generates navigation instructions. The two networks share common latent features, for mutual enhancement through a double back translation model: Generated paths are translated into instructions while generated instructions are translated into path The experimental results show the benefits of our approach in terms of instruction understanding and instruction generation.

I. INTRODUCTION

Domestic service robots (DSRs) are promising solutions for the support of older adults and disabled people. Efforst are increasingly being made to standardize DSRs to provide various support functions [1]. Among these functions, the ability to navigate in an indoor environment is crucial as it is a pre-requisite to many daily life tasks such as fetching a glass of water from the kitchen. However, in the case of most DSRs, the ability to interact through language, while being user-friendly for the non-expert user, is limited by the complexity of understanding natural language.

In this context, we focus on understanding natural language instructions for indoor navigation. This task involves predicting a sequence of actions to reach a goal destination from instructions such as "'*Go down stairs. At the bottom of the stairs walk through the living room and to the right into the bathroom. Stop at the sink.*" Such a task presents several challenges related to the ambiguity of the instructions because the many-to-many nature of mapping between language and the real world makes it difficult to accurately predict user intention. In particular, unlike the understanding of manipulation instruction [2], [3] based on a single environment state, this task requires the language to be mapped to changing states as the DSR moves towards the
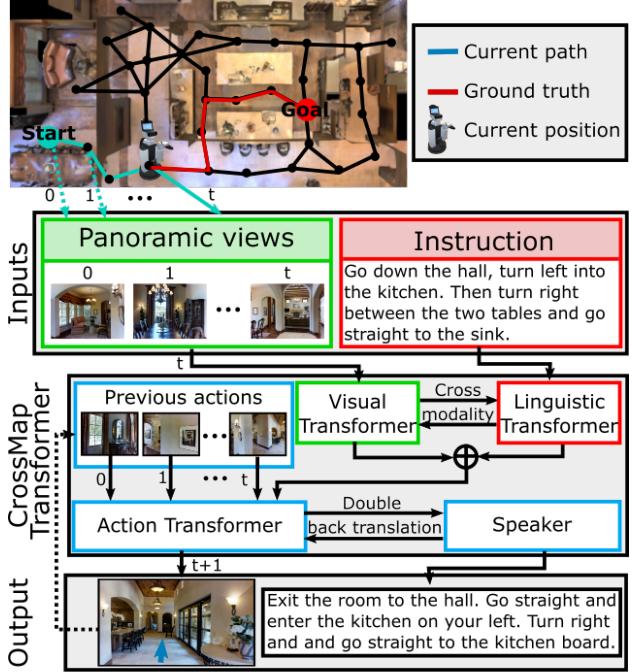

Fig. 1: Our approach, the CrossMap Transformer is used to predict the sequence of actions to navigate to a goal destination given an instruction.

destination. Furthermore, navigation instructions are generally longer than manipulation instructions, which increases the complexity of the task.

The task has recently been formalized as visual-and-language navigation [4] (VLN), and many approaches based on data-driven methods [5], [6] have been proposed. Classically, these approaches exploit a recurrent neural network to infer a sequence of actions leading to the desired destination. Although, these approaches have shown promise, their level of accuracy remains far from that of a person [4].

To narrow this accuracy gap, we propose the Cross-modal Masked Path (CrossMap) Transformer. Motivated by the recent development of transformer networks [7] for language modeling [8], [9], the CrossMap Transformer encodes linguistic and environment state features to sequentially generate actions similarly to recurrent network approaches. Our approach uses feature masking to better model the relationship between the instruction and environment features.

Additionally, we use a double back translation (DBT) approach. Unlike [5] where a speaker network is trained separately to generate instructions, the DBT consists in

[1]The authors are with the National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika, Soraku, Kyoto 619-0289, Japan. `firstname.lastname@nict.go.jp`

[2]The author is with Keio University, 3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa 223-8522, Japan. `firstname.lastname@keio.jp`

mutually training the speaker network, and the CrossMap Transformer, by using common latent features.

The present work makes the following key contributions to the literature:

- We propose the CrossMap Transformer, which sequentially generates actions to reach an instructed destination from linguistic and environment state features. We explain the method in Section IV.
- We propose a double back-translation to improve the mapping between linguistic and action features as explained in Section IV.
- We apply the CrossMap Transformer on the standard Room-to-Room(R2R) dataset [4]. Our approach achieves results comparable to those of recently proposed state-of-the-art recurrent neural network methods. We present the experimental validation in Section V.

II. RELATED WORK

Robot navigation [10] and path planning from natural language instructions have been widely investigated in the field of robotics [11]–[13]. Such a task was recently formalized adopting data-driven methods [4] with the release of the R2R dataset. In this setup, the VLN task [14] is addressed using Long Short Term Memory (LSTM) networks structured in an encoder-decoder framework. An instruction is encoded first and then decoded as a sequence of actions using the current environment states. Initially, the VLN method uses low-level action spaces, where each motion (e.g., left, right or forward motion) of the robot is predicted. The use of a panoramic action space [5] has been shown to improve results as the action sequence directly moves the robot from one position to another. However, the complexity of the VLN task is emphasized by the large gap between human performance and the performance of neural models. Additionally, the trade-off between exploration and instruction fidelity has been emphasized in [15] and additional evaluation metrics, such as the dynamic time warping, have been proposed.

To overcome these limitations, several works [5], [16] proposed exhaustive exploration of the environment. Although these approaches are generally more accurate, they are not feasible in real-world environments. Another line of work relates to data augmentation [17]. Indeed, the R2R dataset is relatively small and introduces several biases in the training set distribution [4]. To mitigate these biases, the Room-for-Room (R4R) dataset was introduced in [15] by synthetically concatenating several paths and instructions. A back-translation model, a speaker [5], has been introduced to generate additional instructions for unlabeled path. This approach was extended in [6] with the environmental dropout, to improve the back-translation model by synthetically generating new environments from dropped features. The exploration of auxiliary tasks [18], such as ensuring the consistency of the path has had positive results. Globally, pre-training on a large dataset has been proven to improve the generalization of VLN models [19], [20].

Our approach, using the CrossMap Transformer, is based on transformer networks. Although such structures have been

Fig. 2: For the VLN task, each environment is given as a navigation graph where each node is a panoramic view and each edge an action to move from one node to another one.

widely used in language modeling [8], [9], [21], few works have used transformers to address the VLN task. In [19], transformers were used to score the compatibility between paths and instructions, in the setting of the pre-exploration of an environment. In [20], transformers were used in pre-training the language embedding model by combining visual and linguistic features. The path was nonetheless predicted from an LSTM-based encoder-decoder. Conversely, [22] proposed to using transformers to decode the instructions and environment features as a sequence of actions. Nonetheless, there remains a gap in performance between this approach and classic LSTM-based networks.

III. PROBLEM STATEMENT

A. Task background

We aim to solve the VLN task in real-world home environments as described in [4]. More specifically, given a natural navigation instruction, this task involves predicting a sequence of actions that leads to a goal destination. This task requires the understanding of instructions such as "Walk out of the bathroom and wait on the stairs that are on the right" and "Go down stairs. At the bottom of the stairs walk through the living room and to the right into the bathroom. Stop at the sink." Such a task is particularly relevant to DSRs, as it complements manipulation tasks [2], [3] for communicative robots that interact naturally with users. Nonetheless, the VLN task is challenging because it requires natural and long sentences to be addressed, where the relationships among individual words and with the current environment state should be modeled.

Additionally, this task requires a trade-off between exploration of the environment and fidelity to the instruction. Indeed, it has been shown that methods based on exploration outperform methods that learn only from the ground truth paths [4]. This is explained by the difference in the distributions of the training and test data when learning only using ground truth methods. Meanwhile, although exploration yields better results, such an approach induces discrepancies between the instruction and generated path [15] which limits the modeling between the instruction and physical world.

Hence, many studies [5], [19] have propose exhaustively exploring the environment and selecting the best path to reach the specified destination. However, considering a physical deployment on DSRs, such methods are cumbersome and time consuming as an exhaustive map should be built
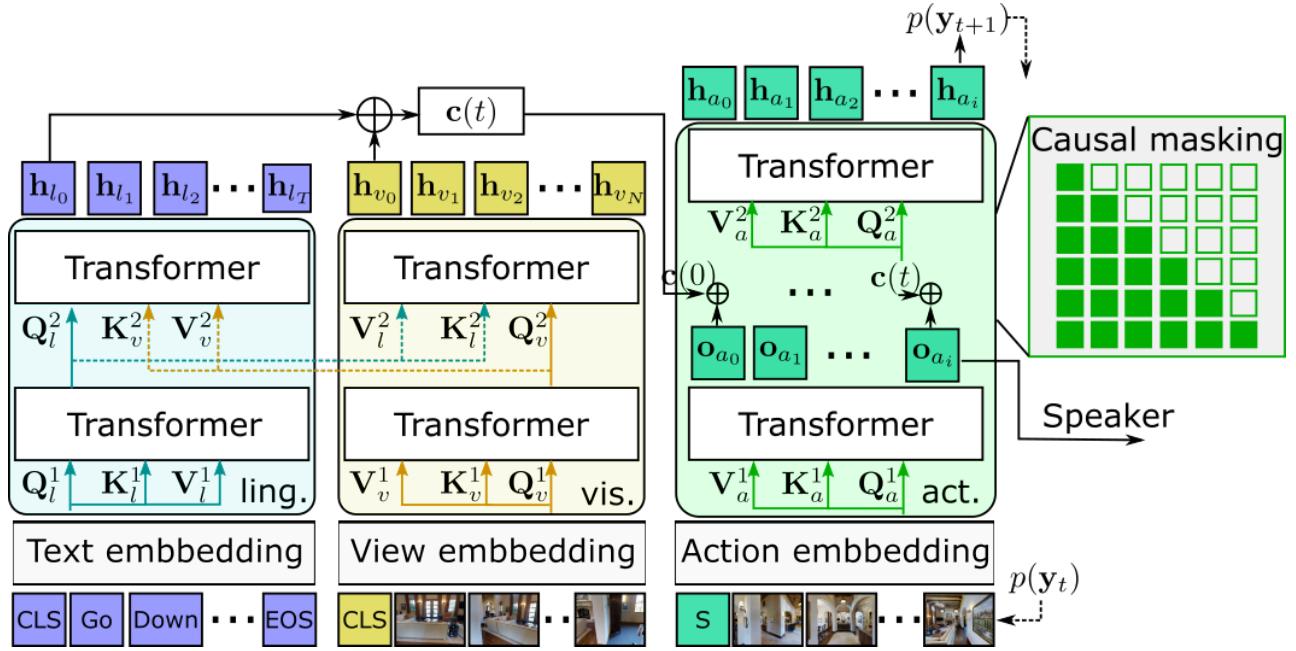
Fig. 3: The CrossMap Transformer (CMT) model endows linguistic, visual and action transformers to predict the sequence of actions given an instruction and a causal masking structure.

and explored before performing the required task. As so, the work, we consider single run approaches, where only one path is generated on-the-fly.

B. Task Description

The task environments of different size are in real home scenes as depicted in Fig. 2. Each scene is discretized into connected nodes that build different paths. A node corresponds to a 360 degree panoramic image.

Similarly to many other works [5], [20], we consider that the VLN is performed through adopting a panoramic action space instead of atomic actions [4] (i.e., atomic motion). Hence,, the navigation task can be represented as a graph where each node represents a waypoint in the scene (see Fig. 2). Each edge of this graph is an action to move from one node to another. A path is then a sequence of nodes from an initial node to a destination node.

Unlike the aforementioned studies [5], [19] that addressed this task through full knowledge of the navigation graph, we consider a single-run setup where only the current node and adjacent edges are known.

In this setup, the VLN task is sequential and for every time step t the following inputs and outputs are expected:

- **Inputs (t=0)**: A navigation instruction as a sentence.
- **Inputs (t ≥ 0)**: The current node (waypoint), the adjacent edges, and a panoramic image taken at the node (waypoint).
- **Output (t ≥ 0)**: Next action (edge) among the adjacent edges.

Four evaluation metrics are considered in this study, that is, the Success Rate (SR) which is the rate of successfully generating a path (arriving within 3 meters of the desired destination), the navigation error (NE), which is the mean distance of arrival from the desired destination, the Success weighted by Path Length (SPL), which is the ratio of successful predictions normalized by the path length, and the oracle success rate (OSR), which is the rate of generated paths that cross within 3 meters of the desired destination.

IV. PROPOSED METHOD

A. Novelty

As explained in the previous section, there is a gap in performance between state-of-the-art methods using the LSTM architecture [6] and those using transformers [22]. Indeed, transformers are generally used for feature modeling [8], [23], achieving state-of-the-art results. Conversely, sequence generation has seldom been addressed [9], and very few architectures are optimized for this type of task.

To narrow this gap, we propose an original transformer architecture for VLN, the CrossMap Transformer (CMT) illustrated in Fig. 3. Our approach predicts a sequence of actions that reaches the specified goal via causal masking and at the same time learns the relationship between the instruction and the corresponding path. Furthermore, we introduce a transformer-based speaker, CrossMap Speaker (CMS) to improve the generalization ability of our approach through a double back-translation (DBT) model. We advocate that a better model can be obtained when mutually training the CMS and CMT networks with common latent features.

The CMT has the following characteristics:

- The CMT combines visual and linguistic modalities to predict a path through a sequence of actions.
- The CMT adopts cross-modal path masking to model the relationship between the navigation path and instruction.

- The CMT and CMS are mutually trained through DBT, which enhances both networks.

B. CrossMap Transformer Architecture

1) Network Inputs: Let us consider an instruction i, such that at each time step t of the sequence, the set of inputs of the network is defined as:

$$\mathbf{x}^i(t) = \{\mathbf{x}_l^i, \mathbf{x}_c^i(t), \mathbf{x}_a^i(t), \}, \tag{1}$$

where $\mathbf{x}_l^i$ and $\mathbf{x}_a^i(t)$ respectively denote the linguistic and previous action inputs and $\mathbf{x}_c(t)$ is the current navigation node. In the following, the index i is omitted for simplicity.

In detail, $\mathbf{x}_l$ is the embedded instruction. The current navigation node is given as a set of N image views so that

$$\mathbf{x}_c(t) = \{I_0(t), I_1(t), \ldots, I_n(t), \ldots, I_N(t)\}, \tag{2}$$

where each image $I_n(t)$ is a portion of the input panoramic image at step t. In this study, we consider that $N = 36$ as proposed in [4].

The previous actions are given as a sequence

$$\mathbf{x}_a(t) = \{E(0), E(1), \ldots, E(t-1)\}, \tag{3}$$

of the $t-1$ steps actions performed. An action $E(t)$ is defined as a given view of the current panorama in the direction of the next navigation node.

2) Language Encoder: Each instruction $\mathbf{x}_l$ is initially tokenized into subwords that are then embedded as 384-size vectors using MiniLM network [21]. MiniLM is the distilled version of UniLM [9], which is a state-of-the-art method of masked language modeling based on transformers. These vectors are then processed through the language encoder. The language encoder is a two-layer transformer that takes as input the tokens features and the corresponding positional encoding. The maximum instruction sequence length is set to 42, that is, 40 tokens and as the beginning-of-sentence (<CLS>) and end-of-sentence (<EOS>) tokens. Each sentence is padded according to its length with the padding token (<PAD>).

In the language encoder, self-attention heads process the input vectors and are followed by a fully connected feed-forward network. The output of an attention head is given by:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}, \tag{4}$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are the queries, keys and values, whereas $\mathbf{M}$ is a mask matrix, controlling where each token can attend. The mask $\mathbf{M}$ takes a value of $0, -\infty$ to allow or prevent attention. Within this transformer, self-attention and linguistic cross-modal attention are applied as defined in Equation (4) by also considering the current environment state through the $\mathbf{x}_c(t)$ to condition the language input. In the self-attention configuration, $\mathbf{M}$ corresponds to a linguistic mask matrix while $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are projections of the transformer hidden vector. For the linguistic cross-modal attention, the queries $\mathbf{Q}$ and values $\mathbf{V}$ are projections of the visual features $\mathbf{x}_c(t)$. The <CLS> is used as the current representation of the instruction and is output as a vector $\mathbf{h}_{l_0}(t)$ of dimension 1×384.
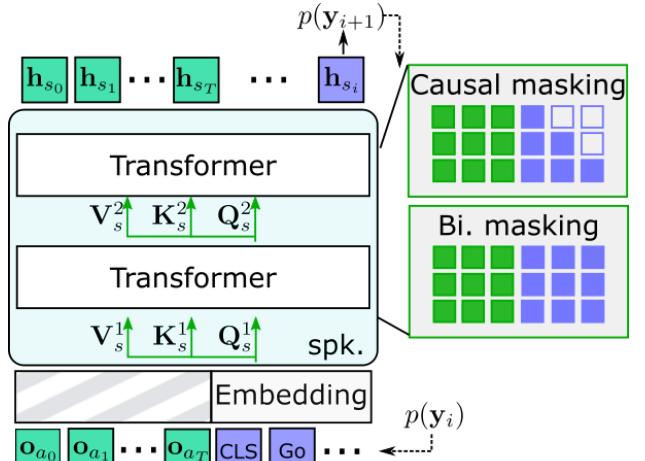

Fig. 4: CrossMap Speaker (CMS) configuration that combines the sequences of latent actions features from the CrossMap Transformer (CMT) and the instructions. The CMS uses both causal and bidirectional masking.

3) Visual Encoder: Each image is encoded as the concatenation of the semantic view provided in [24] and the ResNet-152 [25] features as provided in [4]. The intuition behind this approach is to use different granularity of features. Indeed, ResNet-152 provides low-level features while the semantic features provides to high-level features.

A positional feature is also input to the visual encoder by concatenating each visual feature with the set of features $[\cos(\theta_n), \sin(\theta_n), \cos(\phi_n), \sin(\phi_n)]$ that encodes the relative azimuth θ_n and elevation ϕ_n of a view I_n with respect to the current robot orientation.

Similarly to the linguistic encoder, cross-modal attention is used in the visual encoder. Indeed, following the self-attentive transformer layer, each attention head is conditioned by linguistic features obtained from the linguistic encoder. A visual representation at the step t is obtained using <CLS> token. This token is processed and output as a vector $\mathbf{h}_{v_0}(t)$ of dimension 1×384.

4) Action Decoder: The action decoder predicts the likelihood of all candidate actions given the linguistic and visual features at step t and the previous sequence of actions. Each action is embedded similarly for each panoramic view, that is by concatenating the high-level semantic features and low-level ResNet-152 features. Additionally, two types of positional encoding are used for the action decoder. The first positional encoding corresponds classically to the position of the action in the sequence. For the second type of positional encoding, each action is concatenated with the set of features $[\cos(\theta_m), \sin(\theta_m), \cos(\phi_m), \sin(\phi_m), \rho_m]$ that encodes the relative azimuth θ_m and elevation ϕ_m and distance ρ_m to an adjacent node m with respect to the current robot pose. For the stop action, the azimuth and elevation angles as well as the distance are set to zero.

These inputs are processed by the action decoder. The first layer outputs $\{\mathbf{o}_a(0), \ldots \mathbf{o}_a(t)\}$, which is then concatenated with the corresponding context

feature $\mathbf{C}(t) = \mathbf{h}_{l_0}(t) \bigoplus \mathbf{h}_{v_0}(t)$. The features $\{\mathbf{o}_a(0) \bigoplus \mathbf{C}(0) \ldots \mathbf{o}_a(t) \bigoplus \mathbf{C}(t)\}$ are then processed by the second layer of the transformer. It is noted that the intermediate outputs $\{\mathbf{o}_a(0), \ldots \mathbf{o}_a(t)\}$ are also used for the CMS network. Such an architecture allows training both the CMT and CMS. For the action decoder, a causal masking is used, that is $\mathbf{M}$ corresponds to a mask where each embedded feature attends only the previous features of the sequence. The last output $\mathbf{h}_a(t)$ of the transformer is then used to compute the likelihood $p(\mathbf{y})(t+1)$ given as

$$p(\mathbf{y})(t+1) = \mathbf{h}_l(t)\mathbf{y}(t+1), \tag{5}$$

where $\mathbf{y}$ corresponds to the set of candidate actions from the current robot position.

5) Loss function: To predict the likelihood $p(\mathbf{y}(t))$ of an action, the loss function L as the cross-entropy function is expressed as:

$$L = -\sum_n \sum_m y_{nm}^* \log p(y_{nm}), \tag{6}$$

where y_{nm}^* denotes the label given to the m-th dimension of the n-th sample, and y_{nm} denotes its prediction. As performed in [4], the next action $\mathbf{x}_a(t+1)$ in the sequence is sampled from $p(\mathbf{y}(t))$ to allow the agent to explore the environment.

6) Cross-modal Path Masking: A novelty of the CMt is the cross-modal path masking. In addition to the loss function describe above, the CMT learns the relationship between the ground truth paths and the corresponding instruction. Similarly to NLP methods such as BERT [8], a ground truth sequence of action is randomly masked, and predicted by the network. Given a ground truth sequence of action $\{\mathbf{y}(0), \ldots \mathbf{y}(T-1), \mathbf{y}(T)\}$ of size T, and a masked position m, $m < T$, the CMT computes $p(\{\mathbf{y}(m)|\mathbf{y}(m-1) \ldots \mathbf{y}(0))$ through a causal mask and the same transformer architecture described previously. This approach is used in the pre-training phase.

7) Speaker: In addition to the CMT, we introduce the CMS (see Fig. 4) network that is a two-layer transformer generating a sentence from a sequence of actions. The CMS uses a similar architecture as visual and language transformers such as VLP [26]. The inputs of the CMS are the full sequence of latent action features $\mathbf{o}_a(t)$ that are computed by the CMT. The CMS generates the sequence of words using alternatively causal masking to generate the sequence of words and bi-directional masking to learn the relationship between the instruction and the sequence of actions.

8) Double back-translation: Such an architecture where the CMT and CMS share the same features representation allows $\mathbf{o}_a(t)$ to be concurrently processed into a sequence of actions and translated into an instruction. This approach mutually enhances the CMT and CMS network, differently to the method proposed in [5] where the speaker is trained separately. In addition, more classically, we used a second translation by using the sentences generated by the CMS to train the CMT. We define these two levels of translation as the double back translation (DBT).

To perform the DBT, the latent features $\mathbf{o}_a(t)$, for a full sequence generated by the CMT, are used to train the CMS only if the instructed destination is successfully reached. Against this context, the CMS minimizes the cross-entropy loss between the generated sentence and the original instruction.

In a second phase the CMS generates sentences from latent features $\mathbf{o}_a(t)$ obtained from ground truth paths. These sentences are used to train the CMT. Only sentences with a scoring metric greater than a threshold λ are used.

Finally if the CMS scoring metric is greater than λ on the validation set, the speaker is then used on unlabeled paths to generated sentences and train again the CMT. By filtering both the CMT and CMS, we limit the impact of noisy generated features as analyzed in [27].

V. EXPERIMENTS

A. Experimental Setup

Parameters of the CMT are summarized in Table I.

Each of the transformers comprised two layers of 12 attention heads, with a hidden dimension of 384, while the feed-forward layers had a dimension 1534. We applied a dropout rate of 0.1 to each layer. In the visual encoder, we used the environmental dropout on the visual features with a rate of 0.4.

The training procedure was divided into two phases. First, Both the CMT and CMS were mutually pre-trained using ground truth paths and cross-modal path masking. In the second phase, both networks were fine-tuned following the setup given in Section IV. Unlike previous works, we selected Spice [29] as the CMS scoring metric, and set $\lambda = 20$. It was shown in [27], that this metric is most consistent with human labelling in the VLN for the task.

The CMT was trained on a machine equipped with four Tesla V100 with 32 GB of memory, 768 GB RAM and an Intel Xeon 2.10 GHz processor. The results were reported after 300 epochs. With this setup, it took around 1 day to train the CMT with a batch size of 50 samples and at learning rate of 5×10^{-4}.

TABLE I: Parameter settings and structures of the CrossMap Transformer (CMT)

CMT	Adam (lr= $5e^{-4}$,
Opt. method	$\beta_1 = 0.99$, $\beta_2 = 0.9$)
Nb.layers	2
Hidden size	348
Language Model	MiniLM [21]
Activation	ReLu
Nb heads	12
Feed-forward size	1534
Dropout	0.1
Env. Dropout	0.4
λ (Spice)	20
Batch size	50

TABLE II: Comparison results of the VLN task for LSTM-based networks and Transformer-based networks (Trans.) under several metrics: Success Rate (SR), Navigation Error (NE), Success Path Length (SPL) and Oracle Success Rate (OSR). The results for PREVALENT* are obtained from our own implementation.

Method	Network Type	Validation Seen				Validation Unseen			
		SR↑	NE↓	SPL↑	OSR↑	SR↑	NE↓	SPL↑	OSR↑
Seq2Seq [4]		0.39	6.01	–	0.53	0.22	7.81	–	0.28
Speaker-Follower [5]		0.66	3.36	–	0.74	0.35	6.62	–	0.45
RCM [16]		0.67	3.37	–	0.77	0.43	5.88	–	0.52
EnvDrop [6]	LSTM	0.62	3.99	0.59	–	0.52	5.22	0.48	–
AuxRN [18]		0.70	3.33	0.67	**0.78**	0.55	5.28	0.50	0.62
PREVALENT [20]		0.69	3.67	0.65	–	0.58	4.71	**0.53**	–
PREVALENT*		0.70	3.48	0.67	0.77	0.57	4.66	0.52	**0.65**
CMG-AAL [28]		**0.73**	**2.74**	**0.69**	–	**0.59**	**4.18**	0.51	–
PTA [22]	Trans.	0.66	3.35	**0.64**	0.74	0.43	5.95	0.39	0.49
CMT (Ours)		**0.73**	**2.82**	0.63	**0.80**	**0.55**	**4.60**	**0.44**	**0.63**

B. R2R Dataset

The R2R dataset [4] is based on the Matterport3D [24] environments annotated with navigation instructions. The dataset contains 21,567 instructions, which are 29 words long on average, for 90 different environments. The dataset was split with 14,025 instructions in 60 different environments as the training set. Two different validation sets were considered. The first one, validation seen, contained 1020 instructions for the same environments as the training set. The second one, validation unseen, contained 2349 instructions for 11 environments different from the training set environments.

Additionally, for data augmentation, we used the approx. 170,000 collected paths given in [5]. These paths were collected in the training environments and were initially unlabeled.

C. Quantitative Results

We compared the CMT with state-of-the-art methods in terms of the metrics defined in Section III for the two validation sets of the R2R dataset. It is emphasized that we performed a comparison with approaches that use the same setups as in this study. More explicitly, these setups are the panoramic action space and single run setups, without pre-exploration or beam-search.

The results reported in Table II indicated that our approach obtained results, except for the SPL metric, comparable to those of the currently best performing methods, PREVALENT [20] and CMG-AAL [28]. When compared with another Transformer-based approach such as PTA [22], the CMT achieved better results for all metrics except the SPL in the case of the validation seen dataset. We hypothesize that the lower performance in terms of SPL metric may be related to the fact that we did not optimize the CMT for the shortest path unlike [5], [6]. These works adopt a reinforcement learning approach, which penalizes, among other things, the number of step taken (longer paths). Nonetheless, it is worth mentioning that the SPL scores the optimal (shortest) path but fails to take into account the similarity between the reference and generated trajectory as analyzed in [15].

TABLE III: Ablated results of the CMT for the validation sets. CMT-type1 refers to a model without DBT and path masking, while CMT-type2 is a model without DBT.

Method	Validation Seen			
	SR↑	NE↓	SPL↑	OSR↑
CMT-type1	0.59	4.00	0.51	0.70
CMT-type2	0.64	3.71	0.53	0.71
CMT (Ours)	**0.73**	**2.82**	**0.63**	**0.80**

Method	Validation Unseen			
	SR↑	NE↓	SPL↑	OSR↑
CMT-type1	0.48	5.44	0.35	0.56
CMT-type2	0.50	5.01	0.38	0.55
CMT (ours)	**0.55**	**4.60**	**0.44**	**0.63**

To gain a better insight into the CMT, we performed an ablation study considering the cross-modal path masking, as well as the DBT; results are given in Tables III. We considered two ablation conditions, CMT-type1 that is without DBT and path masking, and CMT-type2 that is without DBT. The results emphasize that both ablated parts greatly improve the performance of the CMT.

D. CrossMap Speaker results

An alternative way to assess the contributions of this study is to evaluate the performance of the CMS network. As stated in Table IV, we evaluated the generated sentences with four captioning metrics that are BLEU-4, ROUGE, CIDEr and SPICE. The CMS is compared with an ablated architecture that is trained separately from the CMT, named CMS-type1, and with the Speaker-follower [5],

Although our approach performed better than the two other baselines only for SPICE metric, our results are consistent with the study [27] that claims that standard captioning metrics, except for SPICE, are ineffective for VLN tasks.

E. Qualitative Results

The qualitative results of the CMT and CMS are illustrated on Fig. 5. Each rows represents a sample instructions from

(a) GT: Enter the bedroom. Turn left and exit the bedroom through the door. Wait by stairs. || CMS: exit the bathroom and turn left. walk past the bed and exit the room. wait there.

(b) GT: Go down the stairs, go slight left at the bottom and go through door, take an immediate left and enter the bathroom, stop just inside in front of the sink. || CMS: go down the stairs and turn left. walk straight and enter the room on the left. stop in front of the sink.

(c) GT: Take a right and walk out of the kitchen. Take a left and wait by the dining room table. || CMS: Walk past the kitchen and turn left. walk past the dining room table and chairs and stop.

(d) GT: Head around the table and go the main area left of the long table. Go to the middle of the room next to the ping pong table and stop. || CMS: Walk past the ping pong table and wait by the ping pong table.

Fig. 5: Qualitative results of the CMT represented as a sequence of actions from an initial panoramic image and a ground truth (GT) instruction. The generated sentences from the CMS are also given.

TABLE IV: Evaluation of generated sentences given the unseen validation set

Method	BLEU-4	CIDEr	ROUGE	SPICE
Speaker [5]	**13.5**	27.2	**33.6**	19.2
CMS-type1	6.4	**27.9**	27.6	20.9
(Ours) CMS	5.3	23.6	27.3	**21.9**

the validation set, and is given as a sequence of actions. The first three samples samples show successfully predicted path from the CMT. The CMS also generated consistent instructions, which suggest that the relation between visual and linguistic features was correctly modeled. The last row illustrates an erroneous sample for both path prediction and instruction generation. Such a sample illustrates one of the challenges of the VLN task, as the table of ping pong, which is a landmark, was seen from far and from several positions. Similarly several tables were in the seen and should be differentiated from each other. These challenges also affected the generated instruction.

VI. Conclusion

In the context of the increasing demand for DSRs, we addressed visual navigation from natural language instruction in home environments. The mainresults of the study are summarized as follows:

- We proposed the CrossMap Transformer, which sequentially generates actions to reach an instructed destination determined from linguistic and visual features.
- We proposed the double back-translation to improve the mapping between linguistic and actions features using a common structure between a CrossMap Speaker network, which translates the sequence of actions into an instruction, and the CrossMap Transformer.
- We achieved state-of-the-art results when using CrossMap Transformer on the R2R dataset.

In future work, we will combine our present achievements with linguistic explanation, allowing interaction with non-expert users in failure cases.

References

[1] L. Iocchi, D. Holz, J. Ruiz-del Solar, K. Sugiura, and T. Van Der Zant, "RoboCup@ Home: Analysis and Results of Evolving Competitions for Domestic and Service Robots," *Artificial Intelligence*, vol. 229, pp. 258–281, 2015.

[2] A. Magassouba, K. Sugiura, and H. Kawai, "A Multimodal Classifier Generative Adversarial Network for Carry-and-Place Tasks From Ambiguous Language Instructions," *IEEE RA-L*, vol. 3, no. 4, pp. 3113–3120, 2018.

[3] A. Magassouba, K. Sugiura, A. Trinh Quoc, and H. Kawai, "Understanding Natural Language Instructions for Fetching Daily Objects Using GAN-Based Multimodal Target-Source Classification," *IEEE RA-L*, vol. 4, no. 4, pp. 3884–3891, 2019.

[4] 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.

[5] 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," in *Advances in Neural Information Processing Systems*, 2018, pp. 3314–3325.

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

[7] 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.

[8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[9] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 063–13 075.

[10] F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. E. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese, "Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 713–720, 2020.

[11] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward Understanding Natural Language Directions," in *ACM/IEEE HRI*, 2010, pp. 259–266.

[12] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy, "Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation," in *AAAI*, 2011, p. 2.

[13] Y.-L. Kuo, B. Katz, and A. Barbu, "Deep compositional robotic planners that follow natural language commands," in *ICRA*, 2020, pp. 4906–4912.

[14] A. Mogadala, M. Kalimuthu, and D. Klakow, "Trends in integration of vision and language research: A survey of tasks, datasets, and methods," *Journal of Artificial Intelligence Research*, 2019.

[15] V. Jain, G. Magalhaes, A. Ku, A. Vaswani, E. Ie, and J. Baldridge, "Stay on the path: Instruction fidelity in vision-and-language navigation," *arXiv preprint arXiv:1905.12255*, 2019.

[16] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6629–6638.

[17] F. Yu, Z. Deng, K. Narasimhan, and O. Russakovsky, "Take the scenic route: Improving generalization in vision-and-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[18] F. Zhu, Y. Zhu, X. Chang, and X. Liang, "Vision-language navigation with self-supervised auxiliary reasoning tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 012–10 022.

[19] 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," *arXiv preprint arXiv:2004.14973*, 2020.

[20] 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.

[21] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *arXiv preprint arXiv:2002.10957*, 2020.

[22] F. Landi, L. Baraldi, M. Cornia, M. Corsini, and R. Cucchiara, "Perceive, transform, and act: Multi-modal attention networks for vision-and-language navigation," *arXiv preprint arXiv:1911.12377*, 2019.

[23] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *European Conference on Computer Vision.* Springer, 2020, pp. 104–120.

[24] 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.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *ECCV.* Springer, 2016, pp. 630–645.

[26] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and vqa." in *AAAI*, 2020, pp. 13 041–13 049.

[27] 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.

[28] 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.

[29] 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.