Cross-modal Semantic Alignment Pre-training for Vision-and-Language Navigation

Siying Wu
University of Science and Technology of China
Hefei, China
wsy315@mail.ustc.edu.cn

Xueyang Fu*
University of Science and Technology of China
Hefei, China
xyfu@ustc.edu.cn

Feng Wu
University of Science and Technology of China
Hefei, China
fengwu@ustc.edu.cn

Zheng-Jun Zha
University of Science and Technology of China
Hefei, China
zhazj@ustc.edu.cn

ABSTRACT

Vision-and-Language Navigation needs an agent to navigate to a target location by progressively grounding and following the relevant instruction conditioning on its memory and current observation. Existing works utilize the cross-modal transformer to pass the message between visual modality and textual modality. However, they are still limited to mining the fine-grained matching between the underlying components of trajectories and instructions. Inspired by the significant progress achieved by large-scale pre-training methods, in this paper, we propose CSAP, a new method of Cross-modal Semantic Alignment Pre-training for Vision-and-Language Navigation. It is designed to learn the alignment from trajectory-instruction pairs through two novel tasks, including trajectory-conditioned masked fragment modeling and contrastive semantic-alignment modeling. Specifically, the trajectory-conditioned masked fragment modeling encourages the agent to extract useful visual information to reconstruct the masked fragment. The contrastive semantic-alignment modeling is designed to align the visual representation with corresponding phrase embeddings. By showing experimental results on the benchmark dataset, we demonstrate that transformer architecture-based navigation agent pre-trained with our proposed CSAP outperforms existing methods on both SR and SPL scores.

CCS CONCEPTS

• **Computing methodologies** → Knowledge representation and reasoning; • **Information systems** → *Information systems applications*;

*Corresponding Author.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548283

KEYWORDS

Vision-and-Language Navigation, Visual-and-Language Pre-training, Semantic Alignment.

ACM Reference Format:
Siying Wu, Xueyang Fu, Feng Wu, and Zheng-Jun Zha. 2022. Cross-modal Semantic Alignment Pre-training for Vision-and-Language Navigation. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548283

1 INTRODUCTION

Vision-and-Language Navigation (VLN) is an emerging and crucial interdisciplinary task at the intersection of computer vision, natural language processing, and artificial intelligence. In the VLN task, the agent is placed in a realistic environment and learns to interpret and carry out the given natural-language instructions to achieve navigation goals. This task will benefit many applications, such as house cleaning, intelligent cruise control, and driver assistance.

In the VLN task, it is important to learn the fine-grained semantic alignment in trajectory-instruction pairs. We present an example in Figure 1 to indicate that a navigation instruction can be effectively grounded onto a trajectory by extracting discriminative semantic phrases from the instruction and properly aligning them with corresponding parts of the trajectory. Such semantic alignment is extremely important in navigation processing. For example, given an instruction, *"Turn around and go towards the bed. Turn left and enter the bathroom. Stop in front of the sink."*, the agent needs to identify which parts of the instruction should be carried out (*e.g., "enter the bathroom"*) for the next moment, which in turn requires the agent to figure out which parts of the instruction have been accomplished (*e.g., "turn around", "go towards the bed", "turn left"*) by aligning them with previously visited scenes.

A variety of approaches have been proposed to address the VLN task [1, 6, 7, 11, 12, 14, 15, 28–30, 32, 42], however, leveraging such fine-grained semantic alignment in trajectory-instruction pairs has not been fully explored. Most existing methods [1, 11] tackle this problem typically by using attentional mechanisms to ground the related words from the given instruction conditioning on the encoded visited observations. However, they rely on RNN to encode visited observations, which is prone to loss of essential information

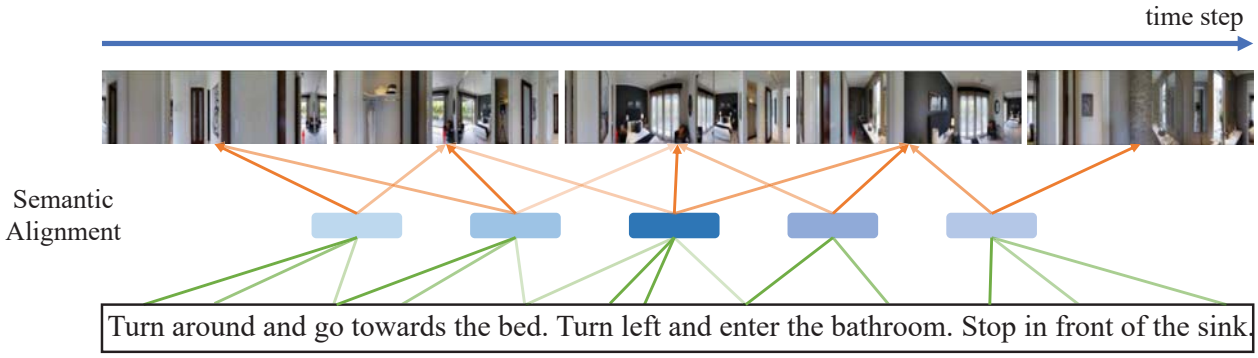

Figure 1: An example of trajectory-instruction pair in R2R dataset. The semantic phrases are extracted from the given instruction, consisting of *"turn around", "go towards the bed", "turn left", "enter the bathroom"* and *"stop in front of the sink"*. These phrases can be aligned to corresponding trajectory segments with same semantic space.

for textual grounding by compressing all histories into a fixed-length vector. Recent works [6, 14, 21] use transformer architecture to store history information and adopt pre-training paradigms to capture the long-range dependencies of the instruction as well as the cross-modal dependencies of the historical observations and instructions. PRESS [21] leverages the large-scale pre-trained language-only model to learn a robust agent. PREVALENT [14] proposes a self-supervised training scheme for a large amount of image-text-action triplets. Airbert [12] proposes BnB, a large-scale in-domain VLN dataset for pre-training transformer-based navigation agents. HAMT [6] first encodes all visited panoramas with a hierarchical multimodal transformer and then incorporates them into decision-making. Although they have achieved state-of-the-art results, the latent cross-modal semantic alignment still lakes systematical exploration.

In recent years, pre-training methods have been widely used in visual language understanding tasks[20, 22, 24–27, 36, 44–46], such as image text retrieval, visual question answering, reference expression grounding, image captioning, video captioning, and so on. Inspired by the significant progress they have made, in this paper, we propose Cross-modal Semantic Alignment Pre-training method (CSAP) for vision-and-language navigation. The CSAP is built upon HAMT [6] and designed to mine underlying semantic alignment between visual modality and textual modality through two novel pre-training tasks: 1) Trajectory-conditioned Masked Fragment Modeling (TMFM), 2) Contrastive Semantic-Alignment Modeling (CSAM). Specifically, the TMFM takes an instruction with a masked fragment (several consecutive tokens) as input and reconstructs this masked fragment conditioned on the useful visual information. The TMFM enforces the model to learn the cross-modal relationships between the unmasked tokens and relevant parts of the trajectory. The CSAM, which is inspired by [33], consists of two modules, a phrase extractor and a semantic aligner. The phrase extractor leverages the linguistic dependencies between words to extract a set of phrases. Many of these phrases are highly semantically similar. Thus, a phrases suppressor is adopted to filter out the discriminative ones by measuring the degree of semantic similarity between them. After that, we apply a semantic aligner to ground the reserved phrases onto the relevant trajectory segments. The CSAM is optimized by contrastive attention loss to facilitate the correct alignment.

The main contributions are summarized as follows:

1) First, we present an effective trajectory-conditioned masked fragment modeling algorithm that learns the relationships between trajectory segments and unmasked semantic phrases, thus enhancing the textual-visual matching.

2) Second, we introduce a contrastive semantic alignment modeling to learn the underlying matching between semantic phrases and trajectory segments for better cross-modal understanding.

3) Third, we conduct extensive experiments to validate the effectiveness of our method and show that it outperforms existing methods on the benchmark dataset.

2 RELATED WORK

Vision-and-Language Navigation. There are many methods that have been proposed for learning to navigate in the realistic environment. Most of them are based on the Recurrent Neural Network (RNN) with attention mechanisms. These methods first ground surrounding observations to instructions, then fed the attended observation features into RNN to encode trajectories at each time step. For example, Anderson *et al.* [3] propose a sequence-to-sequence model to map the language to navigation actions. Early work Speaker-Follower [11] develops a speaker to synthesize new instructions for randomly sampled trajectories and enables the agent with panoramic viewpoints. EnvDrop [38] increases the diversity of synthetic data by randomly removing objects to generate "new environments". Self-monitoring [28] designs a progress monitor and a visual-textual co-grounding module to ensure that the grounded instruction can reflect the navigation progress. Regretful agent [29] proposes a Regret Module that allows the agent to learn when and where to backtrack. Wang *et al.* [42] introduce a novel Reinforced Cross-Modal Matching approach that enforces cross-modal grounding both locally and globally via reinforcement learning. RelGraph [15] designs a language and visual entity relationship graph to exploit the connection among the scenes, objects, and directional clues during navigation. NvEM [1] improves textual-visual matching via adaptively incorporating visual information from neighbor views.

The history plays an important role for an agent to understand its current state. However, RNN based methods are prone to loss of information by compressing all histories into a fixed-length vector. Therefore, a set of methods introduce the graph structure to record history information. Gupta *et al.* [13] construct a top-down belief map of the world and apply a differentiable neural net planner to produce actions. Savinov *et al.* [34] propose semi-parametric topological memory (SPTM) and a deep network capable of retrieving nodes from the graph given observations. Deng *et al.* [8] introduce the Evolving Graphical Planner (EGP) to dynamically build a graphical representation of the environment. Wang *et al.* [41] adopt a similar scene graph to allow an agent to access its past perceptions. Lin *et al.* [23] propose a scene-intuitive agent that knows where to navigate and what objects to locate. PTA [7] introduces a fully-attentive model to achieve interdependency between perception and action. Recent work HAMT[6] efficiently encodes all the past panoramic observations via a hierarchical vision transformer. Our proposed CSAP builds on HAMT with two novelly designed pre-training tasks.

Visual-and-Language Pre-training. The pre-trained BERT model has achieved significant success on a wide range of natural language understanding tasks. After that, the model has been extended to learn the visual-linguistic representations by pre-training on large-scale image-text pairs. ViLT [19] adopts Vision Transformer (ViT) [10] and trains it with associated texts in an end-to-end manner. Lxmert [37] presents a cross-modal transformer framework for learning the connections between vision and language. XGPT [43] proposes a new method of cross-modal generative pre-training for image captioning. A few works research cross-modal pre-training for the VLN task. Hao *et al.* [14] propose a pre-training and fine-tuning paradigm for improving generalization to previously unseen environments on a large amount of image-text-action triplets in a self-supervised learning manner. Majumdar *et al.* [30] pre-train the agent on image-text pairs from the web before fine-tune on embodied path-instruction data. HAMT [6] takes advantage of various proxy tasks to pre-train the hierarchical history encoder. In this paper, we design two proxy tasks to further improve the modeling capability of cross-modal semantic alignment.

3 PRELIMINARY

We firstly review Vision-and-Language Navigation (VLN) task in Section 3.1. Then we introduce the backbone (HAMT [6]) of our algorithm (Figure 2), which consists of a uni-modal encoder in Section 3.2 and a cross-modal encoder in Section 3.3. Our approach based on HAMT is described in Section 4.

3.1 Vision-and-Language Navigation

Vision-and-Language Navigation task requires an agent to navigate from the initial location to the target location following the given natural-language instruction X. The instruction consists of a serious of words, $X = \{x_1, x_2, ..., x_L\}$, where L is the length of the instruction. We enable the agent with panoramic view. At each time step t, the agent perceives a set of images at each viewpoint $V_t = \{v_{t,1}, v_{t,2}, ..., v_{t,K}\}$, where $K = 36$ (12 headings $\times$ 3 elevations with 30 degree intervals) . Each element $v_{t,k} \in V_t$ represents the visual feature of the k-th viewpoint . Following the common practice, we concatenate the visual feature vector $v_{t,k}$ with a 4-dimensional orientation feature $r_{t,k} = [\sin\psi_{t,k}, \cos\psi_{t,k}, \sin\theta_{t,k}, \cos\theta_{t,k}]$, which represents the relative angle to face the view. ψ and θ are the heading and elevation angles respectively. Therefore, the panoramic observation can be denoted by $O_t = \{[v_{t,1}; r_{t,1}], [v_{t,2}; r_{t,2}], ..., [v_{t,K}; r_{t,K}]\}$. Similar to [11], the action space $A_t = \{a_{t,i}\}_{i=1}^{N} = \{[v^c_{t,i}; r^c_{t,i}]\}_{i=1}^{N}$ is the collection of all navigable viewpoints, where N is the number of navigable viewpoints. The agent selects an action from the action space to carry out and finally form a navigation trajectory $\tau = \{(O_1, a_1), (O_2, a_2), ..., (O_T, a_T)\}$, where T is the length of the full trajectory.

3.2 Uni-modal Encoder

Instruction Encoder. The BERT [9] model has shown powerful language modeling capability, which is a multi-layer transformer architecture [40]. We leverage the BERT model to extract contextual word embeddings for the given instruction. Firstly, the instruction is padded to a fixed-length sequence. Then, for each token x_l in the instruction, its representation is a sum of token embedding, position embedding, and type embedding. We add a special token [CLS] at the beginning of the sequence to indicate the start of the instruction. Finally, we adopt the transformer with N_L layers to extract the contextual word embeddings for each token following the standard BERT. The representation of the instruction can be formulated as $X' = \Phi^{BERT}(X) = \{x'_{cls}, x'_1, x'_2, ..., x'_L\}$, where $\Phi^{BERT}(\cdot)$ is the BERT model, X is the input instruction.

Observation Encoder. In the panorama, all viewpoints can be divided into three types, one is the navigable viewpoint that can be visited, the other is the non-navigable viewpoint, and the third is the stop location (the stop token is appended to observation to support stop action). We use $E^T_{v_{t,k}}$ to distinguish different types of the viewpoints. For each viewpoint in the panorama, its representation can be computed as :

$$O'_{t,k} = LayerNorm(W_v v_{t,k}) + LayerNorm(W_r r_{t,k}) + E^T_{v_{t,k}} + E^T_o, \quad (1)$$

where W_v and W_r are learnable weights. E^T_o is the type embedding of the observation encoder. Since the feature dimension of orientation feature $r_{t,k}$ is much lower than that of visual feature $v_{t,k}$, we apply the layer normalization [4] to balance.

Trajectory Encoder. We hierarchically encode the trajectory. First of all, each viewpoint within the panorama is encoded by Equation (1). Then, we stack a transformer with N_h layers to learn the spatial relationships within the panorama and apply average pooling to obtain panorama embedding O'_t. The final temporal token of the trajectory is computed as:

$$\tau'_t = LayerNorm(W_\tau O'_t) + LayerNorm(W_a a_t) + E^P_t + E^T_\tau, \quad (2)$$

where W_τ and W_a are learnable weights. E^P_t is the positional embedding. a_t is the oriented view image feature. E^T_τ is the type embedding of the trajectory encoder. We add a special [CLS] token at the beginning of the trajectory to learn the global representation of the trajectory, which is initialized by a zero vector.

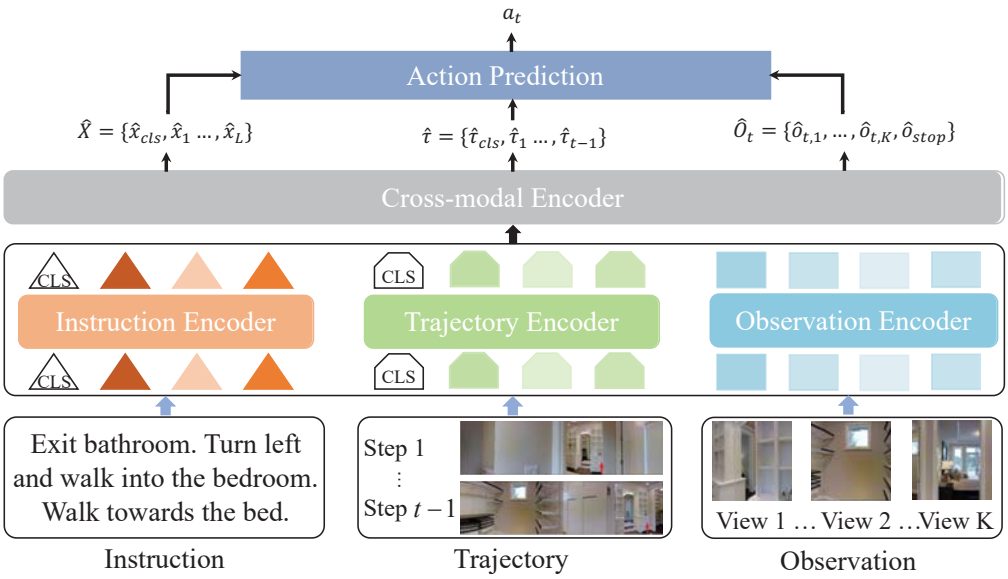

Figure 2: The architecture of our method which consists of an instruction encoder, a trajectory encoder, an observation encoder, and a cross-modal encoder.

3.3 Cross-modal Encoder

We stack N_x cross-modal layers [37] to learn the cross-modal long-range dependencies of trajectories and instructions. Inside each cross-modal layer, the bi-directional cross-attention sub-layer is firstly performed to highlight the relevant visual information for textual modality and relevant textual information for visual modality. Then, a self-attention sub-layer followed by a fully-connected neural network is applied to model the intra-modality relationships for each modality. Different pre-training tasks correspond to different visual inputs. For the given triple input $(X, \tau_{1:(t-1)}, O_t)$, the model takes the concatenation of the trajectory feature and observation feature as input. For the given full trajectory-instruction pair (X, τ), the model takes the only trajectory feature as input. The outputs of the cross-modal encoder are $\hat{X} = \{\hat{x}_{cls}, \hat{x}_1, \hat{x}_2, ..., \hat{x}_L\}$, $\hat{\tau} = \{\hat{\tau}_{cls}, \hat{\tau}_1, \hat{\tau}_2, ..., \hat{\tau}_{t-1}\}$ and $\hat{O}_t = \{\hat{o}_{t,1}, \hat{o}_{t,2}, ..., \hat{o}_{t,K}, \hat{o}_{stop}\}$ for tokens in instruction, trajectory and observation respectively.

4 METHODOLOGY

In this section, we detail the proposed Cross-modal Semantic Alignment Pre-training method (CSAP), including two novel pre-training tasks, Trajectory-conditioned Masked Fragment Modeling (TMFM) in Section 4.1 and Contrastive Semantic Alignment Modeling (CSAM) in Section 4.2. In addition, we detail the training strategy in Section 4.3.

4.1 Trajectory-conditioned Masked Fragment Modeling

Trajectory-conditioned Masked Fragment Modeling (TMFM) aims to learn the fine-grained cross-modal relationships by reconstructing the masked consecutive tokens. This task is similar to the idea of Masked Seq-to-Seq in MASS [35] and Image-conditioned Masked Language Modeling in XGPT [43]. Given a natural-language instruction X, we randomly mask a fragment from position u to v ($0 < u < v < L$). The masked instruction is denoted as $X^{\setminus u:v}$. The number of tokens being masked is k, where $k = v - u + 1$. Each masked token is replaced by a special token [MASK]. Obviously, the length of the instruction is not changed after mask operation. The TMFM is pre-trained to reconstruct the fragment $x^{u:v}$ by taking the full trajectory τ and masked instruction $X^{\setminus u:v}$ as input. The optimal objective is to minimize the negative log-likelihood:

$$L_{TMFM} = -\log P(x^{u:v}|X^{\setminus u:v}, \tau) \tag{3}$$

$$= -\sum_{t=u}^{v} \log P(x_t^{u:v}|x_{<t}^{u:v}, \hat{X}^{\setminus u:v}). \tag{4}$$

Notice that , in the decoder side, the fragment $x_{<t}^{u:v}$ is given when predict $x_t^{u:v}$, $(u \le t \le v)$. The transformer architecture is used in our model as the decoder, considering its state-of-the-art performance on sequence generation tasks. The network architecture is presented in Figure 3.

The Masked Language Modeling in [14] and Trajectory Retelling Task in [47] can be viewed as special cases of the proposed TMFM. When $k = 1$, it becomes Masked Language Modeling. The masked fragment $x^{u:v}$ contains only one token. The decoder predicts the masked words based on the attended images. However, there are many words with no specific visual meaning, the model tends to predict the masked words by the dependencies between unmasked words, rather than by extracting useful visual information. Therefore, the semantic connection of the two modalities can not be sufficiently modeled. When $k = L$, it becomes the Trajectory Retelling Task. All tokens in the instruction are masked. The decoder reconstructs complete instruction according to the full trajectory. The cross-modal encoder cannot learn alignments across modalities

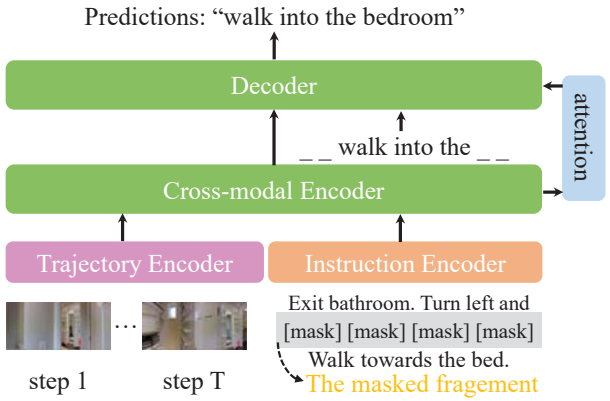

Figure 3: An illustration of the proposed Trajectory-conditioned Masked Fragment Modeling (TMFM).

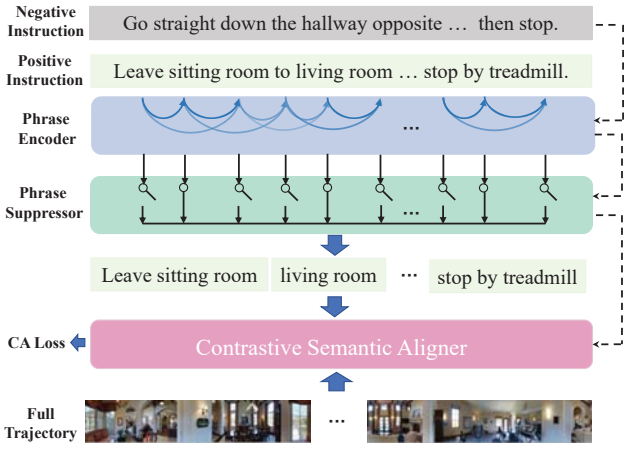

Figure 4: An illustration of the proposed Contrastive Semantic Alignment Modeling (CSAM), which consists of a phrase extractor and a contrastive semantic aligner.

with only visual inputs. In our proposed masked phrase reconstruction task, the agent is forced to extract useful visual information by grounding unmasked tokens onto the trajectory.

4.2 Contrastive Semantic Alignment Modeling

Contrastive Semantic Alignment Modeling (CSAM) aims to learn the underlying alignment between components of trajectories and instructions, which consists of a phrase extractor and a contrastive semantic aligner. The architecture of the CSAM is presented in Figure 4.

Phrase Extractor. In general, phrases are more important than individual words in visual language understanding tasks for several reasons. First, many functional words without specific visual meaning often appear in natural languages, such as *"a"*, *"the"* and so on. Second, phrases tend to have clearer referential meanings than words. For example, the phrase *"the window left to the table"* explicitly specifies the window to the left of the table, while the individual word *"window"* does not specify which window it is. Therefore, we try to extract meaningful phrases from the given instruction and learn the cross-modal matching by aligning them to corresponding trajectory segments. Inspired by [33], the phrase extractor consists of a phrase encoder and a phrase suppressor. The phrase encoder adopts a multi-layer transformer architecture (as described in 3.2) to model the dependencies among words. The function of the phrase suppressor is to filter out discriminative phrases from many similar phrases. Specifically, the phrase encoder produces a phrase embedding matrix and a word attention matrix by taking the embedded instruction as input:

$$\boldsymbol{P}, \boldsymbol{A} = \Phi^{BERT}(X), \tag{5}$$

where $\boldsymbol{P} \in R^{L\times D}$, $\boldsymbol{A} \in R^{L\times L}$. The element $a_{i,j} \in \boldsymbol{A}$ refers to the attention weight of word x_i used in constructing the phrase p_j. Since the phrases are formed using the same set of words, the phrase extractor produces many phrases that are very semantically similar. Therefore, we employ a phrase suppressor to eliminate duplicate phrases with high similarity and keep discriminative phrases. The phrase suppressor computes outer product of the word attention matrix as $\boldsymbol{R} = \boldsymbol{A}(\boldsymbol{A}^T)$ to measure the similarities of all phrase pairs. The element $r_{i,j} \in \boldsymbol{R}$ indicates the degree of similarity between phrase p_i and p_j. In the case where $r_{i,j}$ is greater than a given threshold ϵ, then p_i and p_j are considered to be semantically similar. For two phrases that are semantically similar, the one with higher similarity to the rest of the phrases is considered redundant and should be discarded, while the other is kept. In detail, if $r_{i,j} > \epsilon$ and $\sum_m r_{i,m} > \sum_m r_{j,m}$, then the phrase p_i is discarded and the phrase p_j is reserved. We use M to denote the number of reserved phrases.

Contrastive Semantic Aligner. We first adopt phrase extractor to extract M meaningful phrases from the instruction X. Each trajectory segment is naturally separated with different time stamps. Then, we compute the relevant score $\beta_{i,t}$ for each pair of phrase p_i and trajectory segment $\hat{\tau}_t$ conditioned on their representation vector. The relevant scores indicate the semantic similarities between phrases and trajectory segments. As the common practice to measure the relevant between two vectors, we obtain the scores:

$$\beta_{i,t} = w_s^T \sigma(W_p p_i + H_\tau \hat{\tau}_t + b_s), \tag{6}$$

where w_s^T, W_p, H_τ and b_s are learnable parameters, and σ is an activation function. We adopt contrastive attention loss, proposed in [33], to facilitate the correct semantic alignments between visual modality and textual modality. To this end, trajectory-independent negative instructions are first sampled to provide false candidates to the semantic aligner. We strictly requires that the word repetition rate between positive and negative instructions cannot exceed a fixed threshold γ. The positive relevant score $\beta_{i,t}^{pos}$ and negative relevant score $\beta_{i,t}^{neg}$ are obtained by Equation 6. Then, the softmax function are applied to normalize the relevant scores. We calculate $p_t^{ca} = \sum_{i=1}^{M} \beta_{i,t}^{pos}$, which increases with an increase in positive relevance scores relative to the negative relevance scores. The contrastive attention loss is formulated as:

$$L_{CSAM} = \sum_{(X,\tau)} \sum_{t=1}^{T} (-\log p_t^{ca}).$$

4.3 Training Strategy

The entire agent is trained with two distinct learning paradigms, 1) pre-training with our proposed TMFM and CSAM, as well as several common proxy tasks, and 2) fine-tuning with the mixture of imitation learning and reinforcement learning.

Pre-training with proxy tasks. The model is firstly pre-trained to learn the visual-linguistic representation on proxy tasks [6, 14, 27, 37]. Specifically, given the full trajectory-instruction pairs (X, τ), we apply Masked Language Modeling (MLM), Masked Region Modeling (MRM), Instruction Trajectory Matching (ITM), Trajectory-conditioned Masked Fragment Modeling (TMFM) and Contrastive Semantic Alignment Modeling (CSAM) for pre-training. Given the triplet input $(X, \tau_{1:(t-1)}, O_t)$, we apply Single Action Prediction/Regression (SAP/R) and Spatial Relationship Prediction (SPREL) for pre-training.

Fine-tuning with IR+RL. Following [39], we combine Imitation Learning (IL) and Reinforcement Learning (RL) to fine-tune the model for sequential action prediction. Imitation learning encourages the agent to mimic the teacher action a_t^*, which is denoted as selecting the target viewpoint from all navigable viewpoints. The optimal objective is computed as follows:

$$L_{IL} = \sum_t -\log p_t(a_t^*). \tag{7}$$

However, the agent optimized by IL alone results in overfitting on seen environments. Reinforcement Learning (RL) has been introduced to improve the generalization on unseen environments in [1, 39, 47]. In RL [31], the agent samples an action from the distribution $p_t(a_{t,i})$ and learns from the advantage function A_t [6]. The RL loss is computed as:

$$L_{RL} = \sum_t -\log\left(p_t(a_{t,i})\right) A_t. \tag{8}$$

We mix IL and RL for taking their advantages. The mixed loss is the weighted sum of L_{RL} and L_{IL}:

$$L_{MIX} = L_{RL} + \lambda_{IL} L_{IL}, \tag{9}$$

where λ_{IL} is a coefficient for weighting the IL loss.

5 EXPERIMENTS

5.1 Experimental Settings

R2R Dataset. The Room-to-Room (R2R) dataset is built upon the Matterport3D dataset [5]. It contains 10,800 densely-sampled panoramic RGB-D images of 90 real-world building-scale indoor environments and 7,189 paths sampled from its navigation graphs. The R2R dataset annotates each path with three different ground-truth navigation instructions written by humans, providing 21,567 navigation instructions in total, with an average length of 29 words per instruction. The whole dataset is split into training (61 environments, 14,025 instructions), validation seen (61 environments, 1,020 instructions), validation unseen (11 environments, 2,349 instructions), and test unseen (18 environments, 4,173 instructions) subsets.

R4R Dataset. The Room-for-Room (R4R) dataset [18] is an extended version of R2R, which has longer instructions and trajectories. The dataset is split into train, validation seen, and validation unseen sets.

Evaluation metrics. Several widely-used evaluation metrics are adopted in our experiments for quantitative evaluation: (1) Trajectory Length (TL), the average length of the agent's navigation path. (2) Navigation Error (NE), the agent's mean navigation error in meters. The navigation error is defined as the shortest path distance in the navigation graph between the final position and the goal location. (3) Success Rate (SR), the mean success rate in terms of reaching the goal location. The trajectory is considered to be successful if the distance between the final location and the goal location is less than 3m. (4) Success rate weighted by Path Length (SPL): The agent's Success Rate at reaching the goal location can be improved by exploring more of the environment before committing to a decision. However, in a robotics context, longer trajectories have costs. Therefore, SPL was proposed in [2] to trades-off Success Rate against Trajectory Length. Besides, we leverage several additional metrics to measure the performance on the R4R dataset, including Coverage Weighted by Length Score (CLS) [18], the Normalized Dynamic Time Wrapping (nDTW) [17] and the nDTW weighted by Success Rate (SDTW) [17].

Implementation Details. We set $N_L = 9$ for language transformer, $N_h = 2$ for trajectory encoder, and $N_x = 4$ for cross-modal encoder. In TMFM, we set the length of masked fragment $k = 4$. In CSAM, the threshold ϵ is set to 0.2. The word repetition rate γ between positive and negative instruction is set to 10%. We implemented our model in PyTorch. The model is pre-trained on 4 TeslaV100 GPU with a batch size of 32 for 300k iterations. The learning rate of pre-training is 5e-5. We use the R2R training set and augmented pairs from [14] for training. Then we fine-tune the model on a TitanXP GPU with a batch size of 8 for 100k iterations. The learning rate of fine-tuning is 1e-5. We use the same augmented data as [16] for R2R for a fair comparison. In this stage, the parameters of the uni-modal encoder are fixed. The images are represented by ViT-B/16 features [10]. The visual feature is fixed during both pre-training and fine-tuning. We set $\lambda_{IL} = 0.2$ to balance the IL and RL. In all of our experiments, we use the same hyperparameters for the HAMT [6] baseline and our approach.

5.2 Comparison with SOTA

We compare CSAP with several SOTA methods under the single run setting on both R2R and R4R benchmarks. Note that the VLN mainly focuses on agent's performance on unseen splits, so the performance we reported is based on the model which has the highest SR on the validation unseen split.

In table 1, we provide the results on R2R dataset. Row 1-2 indicate the performance of random and human. Row 3-13 report the performance achieved by Recurrent Neural Network based approaches. Row 14-17 report the performance achieved by transformer-based approaches. Row 18 and 19 provide results of the reproduced HAMT and our method.

Notice that, for fair comparison, Row 18 reports the reproduced results with the released implementation of HAMT [6], which is fine-tuned for action prediction after pre-trained on Masked Language Modeling (MLM), Masked Region Modeling (MRM), Instruction Trajectory Matching (ITM), Single Action Prediction/Regression (SAP/R) and Spatial Relationship Prediction (SPREL) with frozen

Table 1: Performance comparisons with state-of-the-art methods on R2R dataset. † indicates reproduced results.

	Method	Validation-Seen				Validation-Unseen				Test-Unseen			
		TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
1	Ramdom	9.58	9.45	0.16	-	9.77	9.23	16	-	9.89	9.79	13	12
2	Human	-	-	-	-	-	-	-	-	11.85	1.61	86	76
3	Seq2seq [3]	11.33	6.01	39	-	8.39	7.81	22	-	8.13	7.85	20	18
4	SF [11]	-	3.36	66	-	-	6.62	35	-	14.82	6.62	35	28
5	Self-monitoring [28]	-	3.22	67	58	-	5.52	45	32	18.04	5.67	48	35
6	RCM [42]	10.65	3.53	67	-	11.46	6.09	43	-	11.97	6.12	43	38
7	Regretful [29]	-	3.23	69	63	-	5.32	50	41	13.69	5.69	48	40
8	EnvDrop [38]	11.00	3.99	62	59	10.70	5.22	52	48	11.66	5.23	51	47
9	OAAM [32]	10.2	-	65	62	9.95	-	54	50	10.4	-	53	50
10	AuxRN [47]	-	3.33	70	67	-	5.28	55	50	-	5.15	55	51
11	RelGraph [15]	10.13	3.47	67	65	9.99	4.73	57	53	10.29	4.57	55	52
12	NvEM [1]	11.09	3.44	69	65	11.83	4.29	60	55	12.98	4.37	58	54
13	SSM [41]	14.7	3.10	71	62	20.7	4.32	62	45	20.4	4.57	61	46
14	PRESS [21]	10.57	4.39	58	55	10.36	5.28	49	45	10.77	5.49	49	45
15	PREVELENT [14]	10.32	3.67	69	65	10.19	4.71	58	53	10.51	5.30	54	51
16	VLN-BERT [16]	11.13	2.90	72	68	12.01	3.93	63	57	12.35	4.09	63	57
17	HAMT [6]	11.15	2.51	76	72	11.46	2.29	66	61	12.27	3.93	65	60
18	HAMT† [6]	10.84	**2.44**	**76**	**74**	12.62	3.79	64	58	12.83	4.21	**62**	56
19	Ours	11.29	2.80	74	70	12.59	**3.72**	**65**	**59**	13.30	**4.06**	**62**	**57**

ViT visual features. Our approach (Row 19) adds two additional proxy tasks (TMFM and CSAM) for pre-training.

We can observe that our approach (Row 19) provides a 1.6% gain on SR and 1.7% gain on SPL over the HAMT† baseline on validation unseen. In addition, our approach provides a 1.8% gain on SPL over the HAMT† baseline on test unseen. These improvements highlight the benefit of learning the fine-grained semantic alignment between semantic phrases and trajectory segments. Rows 3-13 show the results for state-of-the-art RNN based methods on R2R. Our approach is competitive with these previous state-of-the-art methods across all metrics on validation unseen and test unseen. We achieve these significant progress from pre-training on both common proxy tasks and our proposed CSAP.

In Table 2, we also provide the performance comparison to the HAMT baseline on the R4R dataset. Since both trajectories and instructions in the R4R dataset are longer, we adopt the setting in HAMT [6] to remove the text-to-vision cross-attention layer in the cross-modal encoder. On R4R, our approach achieves 2.1% and 1.9% improvements on SR and SDTW over the HAMT† baseline, respectively.

5.3 Ablation Study

We conduct ablation experiments over different components of CSAP on the R2R dataset. Specifically, we study how the Trajectory-conditioned Masked Fragment Modeling (TMFM) and Contrastive Semantic Alignment Modeling (CSAM) contribute to navigation. Table 3 reports the evaluation results on R2R dataset in validation unseen split. In particular, the HAMT† is pre-trained without TMFM and CSAM. The TMFM is a self-supervised proxy task that forces the model to learn the relationships between trajectory segments

Table 2: Performance comparisons with state-of-the-art methods on R4R dataset in val unseen split. † indicates reproduced results.

Method	NE↓	SR↑	CLS↑	nDTW↑	SDTW↑
SF [11]	8.47	24	30	-	-
RCM [42]	-	29	35	30	13
RelGraph [15]	7.43	36	41	47	34
VLN-BERT [16]	6.67	43.6	51.4	45.1	29.9
HAMT [6]	6.09	44.6	57.7	50.3	31.8
HAMT† [6]	**6.16**	42.05	**59.02**	**52.06**	30.94
Ours	6.21	**42.95**	58.62	51.88	**31.53**

Table 3: Ablation study on R2R dataset in val unseen split. † indicates reproduced results.

Method	TMFM	CSAM	TL	NE↓	SR↑	SPL↑
HAMT†	×	×	12.62	3.79	64.3	58.2
	√	×	11.99	3.73	64.9	59.3
	×	√	12.42	3.75	64.4	58.5
Ours	√	√	12.59	3.72	65.3	59.2

Instruction: Head towards the sink. Turn to the right and walk towards the table. Turn and walk behind the chair on the left side of the table. Stop behind the chair looking out at the backyard.

(a) Predicted trajectory by CSAP (succeed). (b) Predicted trajectory by HAMT (failed).

Figure 5: Examples in R2R val unseen split. The HAMT misunderstands the phrase *"head towards the sink"* at the very beginning. While our CSAP successfully navigates to the target location.

and unmasked tokens. The results on Row 2 indicate the effectiveness of the TMFM with a 0.9% gain on SR and a 1.9% gain on SPL. The CSAM uses contrastive attention loss to align the extracted semantic phrases with trajectory segments for better fine-grained cross-modal matching. The model pre-trained with CSAM (Row 3) outperforms the baseline model across all metrics. Row 4 reports the results achieved by our CSAP. We can observe that the CSAP achieves 1.6% and 1.7% improvements over HAMT† on SR and SPL, respectively.

5.4 Qualitative Analysis

To better demonstrate the effectiveness of our proposed CSAP, we present qualitative visualizations of trajectories generated by HAMT† and CSAP under the same environment and instruction in Figure 5. Given the instruction, *"Head towards the sink. Turn to the right and walk towards the table. Turn and walk behind the chair on the left side of the table. Stop behind the chair looking out at the backyard."*, our CSAP correctly aligns the phrase *"head towards the sink"* with the most relevant navigable viewpoint. However, the HAMT† fails the navigation at the beginning for losing *"the sink"* .

6 CONCLUSIONS

In this paper, we propose a new Cross-modal Semantic Alignment Pre-training (CSAP) for the task of Vision-and-Language Navigation. The CSAP approach consists of two novel pre-training tasks, Trajectory-conditioned Masked Fragment Modeling (TMFM) and Contrastive Semantic Alignment Modeling (CSAM). The TMFM learns to reconstruct the masked fragment under the guidance of underlying semantic alignment between the trajectory and unmasked instructions. The CSAM utilizes a phrase extractor to extract several discriminative phrases from the instruction and align them with the corresponding trajectory segments. The extensive experimental results demonstrate that the transformer-based agent pre-trained with our proposed CSAP outperforms multiple state-of-the-art methods.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China under Grant 2020AAA0105702, the National Natural Science Foundation of China (NSFC) under Grants U19B2038 and 61901433, the University Synergy Innovation Program of Anhui Province under Grants GXXT-2019-025, the Fundamental Research Funds for the Central Universities under Grant WK2100000024, and the USTC Research Funds of the Double First-Class Initiative under Grant YD2100002003.

REFERENCES

[1] Dong An, Yuankai Qi, Yan Huang, Qi Wu, Liang Wang, and Tieniu Tan. 2021. Neighbor-view enhanced model for vision and language navigation. In *International Conference on Multimedia*. 5101–5109.

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

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Conference on Computer Vision and Pattern Recognition*. 3674–3683.

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

[5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *International Conference on 3D Vision*. 667–676.

[6] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. 2021. History aware multimodal transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems* 34 (2021), 5834–5847.

[7] Federico Landi Lorenzo Baraldi Marcella Cornia and Massimiliano Corsini Rita Cucchiara. 2019. Perceive, transform, and act: Multi-modal attention networks for vision-and-language navigation. (2019).

[8] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. 2020. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *Advances in Neural Information Processing Systems* 33 (2020), 20660–20672.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[11] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. Speaker-follower models for vision-and-language navigation. *Advances in Neural Information Processing Systems* 31.

[12] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. 2021. Airbert: In-domain pretraining for vision-and-language navigation. In *International Conference on Computer Vision*. 1634–1643.

[13] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. 2017. Cognitive mapping and planning for visual navigation. In *Conference on Computer Vision and Pattern Recognition*. 2616–2625.

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

[15] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. 2020. Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems* 33 (2020), 7685–7696.

[16] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. Vln bert: A recurrent vision-and-language bert for navigation. In *Conference on Computer Vision and Pattern Recognition*. 1643–1653.

[17] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446* (2019).

[18] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. *arXiv preprint arXiv:1905.12255* (2019).

[19] Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*. 5583–5594.

[20] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI Conference on Artificial Intelligence*, Vol. 34. 11336–11344.

[21] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi. 2019. Robust navigation with language pretraining and stochastic sampling. *arXiv preprint arXiv:1909.02244* (2019).

[22] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*. 121–137.

[23] Xiangru Lin, Guanbin Li, and Yizhou Yu. 2021. Scene-intuitive agent for remote embodied visual grounding. In *Conference on Computer Vision and Pattern Recognition*. 7036–7045.

[24] Daqing Liu, Zheng-Jun Zha, Hanwang Zhang, Yongdong Zhang, and Feng Wu. 2018. Context-aware visual policy network for sequence-level image captioning. In *ACM International Conference on Multimedia*. 1416–1424.

[25] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. 2019. Learning to assemble neural module tree networks for visual grounding. In *International Conference on Computer Vision*. 4673–4682.

[26] Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Dechao Meng, and Qingming Huang. 2019. Adaptive reconstruction network for weakly supervised referring expression grounding. In *International Conference on Computer Vision*. 2611–2620.

[27] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems* 32 (2019).

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

[29] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. 2019. The regretful agent: Heuristic-aided navigation through progress estimation. In *Conference on Computer Vision and Pattern Recognition*. 6732–6740.

[30] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. 2020. Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision*. Springer, 259–274.

[31] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. 1928–1937.

[32] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. 2020. Object-and-action aware model for visual language navigation. In *European Conference on Computer Vision*. Springer, 303–317.

[33] Hobin Ryu, Sunghun Kang, Haeyong Kang, and Chang D Yoo. 2021. Semantic grouping network for video captioning. In *AAAI Conference on Artificial Intelligence*, Vol. 35. 2514–2522.

[34] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. 2018. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653* (2018).

[35] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450* (2019).

[36] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530* (2019).

[37] Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490* (2019).

[38] Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to Navigate Unseen Environments: Back Translation with Environmental Dropout. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2610–2621.

[39] Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to Navigate Unseen Environments: Back Translation with Environmental Dropout. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2610–2621.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[41] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. 2021. Structured scene memory for vision-language navigation. In *Conference on Computer Vision and Pattern Recognition*. 8455–8464.

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

[43] Qiaolin Xia, Haoyang Huang, Nan Duan, Dongdong Zhang, Lei Ji, Zhifang Sui, Edward Cui, Taroon Bharti, and Ming Zhou. 2021. Xgpt: Cross-modal generative pre-training for image captioning. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 786–797.

[44] Zheng-Jun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. 2019. Context-aware visual policy network for fine-grained image captioning. *IEEE transactions on pattern analysis and machine intelligence* (2019).

[45] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. 2020. Object relational graph with teacher-recommended learning for video captioning. In *Conference on Computer Vision and Pattern Recognition*. 13278–13288.

[46] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *AAAI Conference on Artificial Intelligence*, Vol. 34. 13041–13049.

[47] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. 2020. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Conference on Computer Vision and Pattern Recognition*. 10012–10022.