Anticipating the Unseen Discrepancy for Vision and Language Navigation

Yujie Lu*
UC Santa Barbara

Huilang Zhang*
McGill University

Ping Nie
Peking University

Weixi Feng
UC Santa Barbara

Wenda Xu
UC Santa Barbara

Xin Wang
UC Santa Cruz

William Wang
UC Santa Barbara

Abstract

Vision-Language Navigation requires the agent to follow natural language instructions to reach a specific target. The large discrepancy between seen and unseen environments makes it challenging for the agent to generalize well. Previous studies propose data augmentation methods to mitigate the data bias explicitly or implicitly and provide improvements in generalization. However, they try to memorize augmented trajectories and ignore the distribution shifts under unseen environments at test time. In this paper, we propose an Unseen Discrepancy Anticipating VISion and Language Navigation (DAVIS) that learns to generalize to unseen environments via encouraging test-time visual consistency. Specifically, we devise: 1) a semi-supervised framework DAVIS that leverages visual consistency signals across similar semantic observations. 2) a two-stage learning procedure that encourages adaptation to test-time distribution. The framework enhances the basic mixture of imitation and reinforcement learning with Momentum Contrast to encourage stable decision-making on similar observations under a joint training stage and a test-time adaptation stage. Extensive experiments show that DAVIS achieves model-agnostic improvement over previous state-of-the-art VLN baselines on R2R and RxR benchmarks.1

1Our source code and data are in supplemental materials.

1 Introduction

Vision-and-language navigation (VLN) tasks have attracted increasing research interests and achieved significant improvements with the emergence of deep learning techniques. VLN is a complex system that requires decision-making conditioned on visually grounded language understanding. There are some unique challenges in the VLN task, such as reasoning over cross-modal input and generalizing to unseen environments.

Previous studies (Wang et al., 2019a; Tan et al., 2019; Shen et al., 2021; Hong et al., 2021) addressed these issues by proposing to enforce cross-modal grounding via imitation learning and reinforcement learning. To improve the generalizability, most prior studies (Fu et al., 2020; Majumdar et al., 2020; Liu et al., 2021; Parvaneh et al., 2020) propose diversified forms of augmentations on input data. Other studies (Li et al., 2019a; Zhu et al., 2020a) explore the utilization of self-supervision from the data or prior knowledge from the pre-trained models. These studies mainly focused on designing techniques that respect training time generalization. However, due to the distribution shift at test time, these studies suffer from the difficulty of generalizing and maintaining robustness under this situation. For instance, test-time decision-making should be invariant to the changes of irrelevant objects in the environment and the changes of the viewpoint. The robustness of vision-and-language agents against these test-time shifts requires more research focus in this area.

In this paper, we study test-time visual consistency (NNCVLN) in the context of VLN. We also demonstrate that a robustly adapted VLN agent at test-time with self-supervision can outperform prior approaches, which suffer from the biased training data distribution. We propose a test-time robust vision-and-language navigation framework, which consists of self-supervised and supervised modules for both imitation and reinforcement learning. We adopt two general architectures for the supervised module, sequence-to-sequence, and transformer-based, which can translate human instructions grounded in the visual environment to history contextualized action sequences. For the self-supervised module, we utilize contrastive learning to encourage the agent to learn the visual consistency and invariant features from the observations. We apply instance-level augmentation methods for visual environment input to gen-
Environment
Turn left and walk down the hallway. When you reach the black chair at the wall, turn left. Once you turn left walk forward and enter the next room on your left. Stop once you are behind the first yellow chair.

Figure 1: Overall Architecture. DAVIS consists of supervised $\theta_{ML}$ and self-supervised $\theta_{CL}$. $\psi$ and $\phi$ provide Imitation Learning $L_{IL}$ and Reinforcement Learning $L_{RL}$ objectives, respectively.

2 Methodology

2.1 Problem Definition

The vision-language navigation task requires an agent to follow the instruction $I$ in a photo-realistic environment $E$. The panoramic view $o_t$ is divided into 36 single views $\{o_{t,i}\}_{i=1}^{36}$, are provided at each time step $t$. The agent start from the viewpoint $S$ and make actions decisions by selecting from the navigable viewpoints with the policy network $\pi$. After navigating to the new viewpoint, the agent observes the new panoramic view. Finally, the agent stop at the predicted target position $T$ and give the navigation route $R$.

2.2 Overall Architecture

The overall architecture of DAVIS is shown in Figure 1. Given the instruction $I$ and panoramic view $o_t$ extracted from photo-realistic environment $E$ at time step $t$, the navigator $N$ predicts action $a_t$, which is applied to $E$ to generate trajectory path $P$. The speaker $S$ learns to translate the $P$ to $\hat{I}$ maximally similar to $I$. The proposed DAVIS is composed of two parts: 1) Supervised part $\psi_{ML}$ (IL) and $\phi_{ML}$ (RL). 2) Self-supervised part $\psi_{CL}$ and $\phi_{CL}$. $\psi_{ML}$ learns to imitate the teacher actions $\hat{a}_t$ at each time step $t$. $\phi_{ML}$ learns to maximize the reward return through the trajectory path. $\psi_{CL}$ learns to maximize the similarity of the actions erate positive samples. To obtain negative samples, we select the temporal different visual input from the same path. Such consistency is then learned via encouraging the agent to predict similar action distributions over positive instruction-observation pairs and dissimilar ones otherwise. To ensure the agent adapted to the distribution shift, we devise the two-stage training strategy. First, the agent is trained with the semi-supervised objectives by encouraging the predictions to obtain minimized entropy across augmentations over train data. Second, the agent utilizes the self-supervised objectives by encouraging minimized entropy across augmentations over test data and updating the framework before inference. Our Test-time Visual Consistency (DAVIS) framework is a general learning paradigm that can be easily applied to existing VLN baselines to boost their performance of generalization to new scenes. Extensive experiments show that DAVIS achieves model-agnostic improvement over previous state-of-the-art VLN baselines on R2R (Anderson et al., 2018) and RxR (Ku et al., 2020) Benchmarks. In summary, our contributions are three-fold:

- We propose a semi-supervised VLN framework DAVIS that enforces stable decision-making via visual consistency regularization.
- To improve generalizability to unseen environments and maintain robustness under distribution shift, we devise a two-stage training strategy for the VLN agent, consisting of semi-supervised training and self-supervised test-time adaptation.
- Empirically, the model-agnostic DAVIS achieves consistent performance gain over state-of-the-art VLN baselines disregarding the architectures over R2R and RxR Benchmarks.
across similar visual observation at each time step $t$. $\phi_{CL}$ learns to maximize the similarity of reward return through similar trajectory paths. We devise a two-stage training strategy for DAVIS: 1) At the training stage, both supervised and self-supervised parts are jointly trained. 2) At test-time, the supervised part is fixed, while the self-supervised continue to adapt.

### 2.3 Model

**Grounded Navigation Reasoning** Our model design is agnostic to different architectures of navigation reasoning grounded in language and vision understanding. Hence, we select two typical navigation reasoning architectures that conditioned on natural language instruction and visual environment observation as our baselines, *a*) Sequence to Sequence. *b*) Transformer-based. For sequence-to-sequence architecture, the encoder-decoder model with a bi-directional LSTM-RNN encoder and an attentive LSTM-RNN decoder are adopted following EnvDrop (Tan et al., 2019). The instruction-to-navigation translation is computed, first by calculating the attentive visual feature at each decoding step $t$ as:

$$\hat{f}_t = \sum_i \text{softmax}_i (f^T_{t,i} W_f \hat{h}_{t-1}) f_{t,i}$$  \hspace{1cm} (1)

The instruction-aware hidden output $\hat{h}_t$ is computed with the combination of hidden output of the LSTM and the attentive instruction feature.

$$\hat{h}_t = \tanh(W[\hat{u}_t; LSTM(\{\hat{f}_t; \hat{a}_{t-1}, \hat{h}_{t-1})])$$  \hspace{1cm} (2)

where $\hat{u}_t$ is the attentive instruction feature, $\hat{a}_{t-1}$ is the previous action. For transformer-based architecture, we follow VLN$^C$BERT to process the language. We get initial state token $s_0$ and tokens $L^I = u_I$ for instruction $I = u_I$ by:

$$s_0, L^I = E_{\text{language}}([\text{CLS}], I, [\text{SEP}])$$  \hspace{1cm} (3)

where $E_{\text{language}}$ is the language encoder of the BERT. $[\text{CLS}]$ and $[\text{SEP}]$ are predefined tokens in the BERT model. Here, we use $[\text{CLS}]$ to represent the initial state. The visual tokens $V_t$ will refer to the language token $L$ as keys and values for observation and instruction attention at each navigation step $t$. For each raw observation $o_t^{raw}$ at time step $t$, it is projected to the textual space of the BERT embedding. And then the final input $X_t$ at the current time step is the concatenation of the state token $s_t$, textual and visual tokens:

$$V_t^o = I_t W_I, X_t = [s_t, L^I_t, V_t^o]$$  \hspace{1cm} (4)

Then the state will be represented as the summary of both textual and visual tokens of all the previous observations, history action sequences. The raw textual and visual tokens are matched with state and language attention weights, which is computed by averaging the scores from $K$ attention heads as:

$$\hat{A}_t = \text{softmax}(\hat{A}_t) = \text{softmax}(\frac{1}{K} \sum_{k=1}^K Q_{lk} K_{l,k})$$  \hspace{1cm} (5)

where $Q_{lk}$ and $K_{l,k}$ represent the state tokens as query matrix and instruction tokens as the key matrix at head $k$ of final layer $l$.

**Data Augmentation** Navigation in the unseen environment requires a model learning invariances across diversified observations. For example, the viewpoint of the camera and the irrelevant objects changes should not influence the agent’s decision-making process. Data augmentations are usually utilized to encourage the model to encode the invariant features to achieve such generalization. To guarantee that the model generalizes at test time under these unseen invariances, we augment the sample at test time and encourage the model to respect invariances and thus maintain robustness even with distribution shift. For each step $t$, we

---

**Algorithm 1** Test-time Visual Consistency

**Require:**
- Train dataset $D_{train}$, validation dataset $D_{val}$ and test dataset $D_{test}$;
- Supervised module parameters $\theta_{ML}$ consists of $\phi_{ML}$ and $\psi_{ML}$;
- Self-supervised consistency module parameters $\theta_{CL}$ consists of $\phi_{CL}$ and $\psi_{CL}$;
- Augmentation Function Pool $A$;

**Ensure:**

1: for each iteration do
2:  for each $N$ in $D_{train}$ do
3:    Sample augmentation function $F_a$ from Pool $A$, apply Augmentation $F_a$, $o' = f(o)$;
4:  Compute loss
5:  $\theta_{ML} = \theta_{ML}^{-1} + \text{update}$, $\theta_{CL} = \theta_{CL}^{-1} + \text{update}$
6:  end for
7: end for
8: for each test sample in $D_{test}$ do
9:  Sample augmentation function $F_a$ from Pool $A$, apply Augmentation $F_a$, $o' = f(o)$;
10:  Compute loss
11:  $\theta_{CL} = \theta_{CL}^{-1} + \text{update}$
12: end for
13: Inference;
We follow MoCo (He et al., 2020) to use the average
θ which measures agreement between raw observa-
OvDrop (Tan et al., 2019) and feature-level modifica-
tion critic (Haarnoja et al., 2018), an off-policy RL
Learning Module. Specifically, we apply soft ac-
state comparison as a supervised Reinforcement
update the policy model given the ground truth
a reinforcement learning method to roll out and
Consistent Soft Actor Critic
Typically we have
decisions
sistency during teacher forcing. The policy
consider taking augmentations for introducing con-
entropy loss for each decision. Furthermore, we
following teacher actions and calculates a cross-
structions to action sequences. Specifically, the
Consistent Teacher Forcing
Typically, we have
momentum coefficient.
that is updated by back-propagation, and
m momentum encoder. The parameters
θ of the momentum encoder is updated as:
θk ← mθk + (1 − m)θq, m ∈ [0, 1] (7)
where θq is the parameters of the query encoder
that is updated by back-propagation, and m is a
momentum coefficient.
Consistent Teacher Forcing
Typically, we have
the Teacher Forcing technique to translate with
a supervised Imitation Learning Module from in-
structions to action sequences. Specifically, the
agent navigates on the ground-truth trajectory by
following teacher actions and calculates a cross-
entropy loss for each decision. Furthermore, we
consider taking augmentations for introducing con-
istency during teacher forcing. The policy ψCL
network learn to maximize the agreement of action
decisions at and ˆat over positive observation pairs.
Consistent Soft Actor Critic
Typically we have a
reinforcement learning method to roll out and
update the policy model given the ground truth
state comparison as a supervised Reinforcement
Learning Module. Specifically, we apply soft ac-
tor critic (Haarnoja et al., 2018), an off-policy RL
algorithm that optimizes a stochastic policy for
maximizing the expected trajectory returns. We
consider taking augmentations for introducing con-
sistency during actor critic learning. Similar to the
query-key encoder learning in visual encoder, we
learn the momentum update for the key critic φk:
φk ← mφk + (1 − m)φq, m ∈ [0, 1] (8)
where φq is the parameters of the query critic and
updated by back-propagation, m is a momentum
coefficient. The learning objective is described in
Section 2.4.
Action Prediction
The output embeddings zv l,t go through a single fully-connected layer to predict
agent actions ˆa1:T. During training, the input is
the sequence of observation-language paired data
and ground truth action sequence. During testing
at timestep t, we input visual observations v1:t and
previous actions ˆa1:t−1 taken by the agent. After
model adaptation during test-time, we select the
action predicted for the last time step ˆat. And then
apply ˆat to the environment which generates the next
visual observation v t+1. Specifically, for sequence-
to-sequence based architecture, the decision will
be made by pt(a t,k) = softmaxk(gT t,k WCG ˆh t). For
transformer-based architecture, the decision will
be made by pt t = ˆA t
v
.

2.4 Learning
We learn a generalizable policy that handles the
distribution shift between train and test data. The
learning procedure is devised in two phase: 1) Semi-
supervised learning on train data split. 2) Self-
supervised adaptation on test-time data. θ CL and
θ RL is jointly learned with the full objectives ag-
gregated by L1CL, L1RL, L1L and LRL at standard
training stage. At test-time adaptation, θ ML is fixed
while θ CL is updated before inference. The learn-
ing procedure is summarized in Algorithm 1.

![Figure 2: Training and Test-time Procedure](image)
Standard Training The standard training procedure is shown in Figure 2. For the imitation learning, the objective is as:
\[
\mathcal{L}_{i}^{IL} = \sum_{t} \mathcal{L}_{i}^{il} = \sum_{t} -a_{t}^{*} \log \pi_{t}(a_{t})
\]  
(9)

For the reinforcement learning, the training objective is:
\[
\mathcal{L}_{RL}^{IL}(\phi, B) = E_{t, B}[\{(Q_{\phi}(o, a) - (r + \gamma(1-d)T))^{2}]  
\]  
(10)

where \( t = (o, a, o', r, d) \) is a tuple with observation \( o \), action \( a \), reward \( r \) and done signal \( d \), \( B \) is the replay buffer, and \( T \) is the target, defined as:
\[
T = (\min Q_{\phi}^{*}(o', a') - \alpha \log \pi_{\phi}(a'|o'))
\]  
(11)

where \( Q_{\phi}^{*} \) is the exponential moving average of the parameters of \( Q_{\phi} \) to improve training stability in off-policy RL algorithms. \( \alpha \) determines the priority of the entropy maximization over value function optimization. The critic is trained by maximizing the expected return of its actions as:
\[
\mathcal{L}(\phi) = E_{a}[\{(T - Q_{\phi}(o, a)) - \alpha \log \pi_{\phi}(a|o)]
\]  
(12)

where actions are sampled stochastically from the policy. Then we aggregate these two objectives as our supervised objective \( \mathcal{L}_{ML} \):
\[
\mathcal{L}_{ML} = \mathcal{L}_{RL}^{IL} + \lambda \mathcal{L}_{il}^{IL} 
\]  
(13)

where \( \lambda \) balances the weight of Imitation Learning objective and Reinforcement Learning objective.

To leverage the information from data, we add self-supervised objectives during navigation. The contrastive teacher-forcing module learns with the objective:
\[
\mathcal{L}_{CL}^{IL} = \log \frac{\exp(q^{T}W_{k+})}{\exp(q^{T}W_{k+}) + \sum_{i=0}^{K-1} \exp(q^{T}W_{k})}
\]  
(14)

The contrastive soft actor critic module learns with the objective:
\[
\mathcal{L}_{CL}^{RL} = \log \frac{\exp(q^{T}W_{k+})}{\exp(q^{T}W_{k+}) + \sum_{i=0}^{K-1} \exp(q^{T}W_{k})}
\]  
(15)

Please refer to Appendix A.1 for the details of the contrastive module.

Finally, we aggregate above two objectives for self-supervised training objective:
\[
\mathcal{L} = \mathcal{L}_{ML} + \lambda \mathcal{L}_{CL} 
\]  
(16)

Test-time Adaptation The test-time procedure is shown in Figure 2 with separate detailed description in Figure 5(b). At test time, we remove the supervised objective and adapt the model with only the self-supervise objective on test-time sample as:
\[
\mathcal{L}_{CE}(\theta; x) = \frac{1}{B} \sum_{i=1}^{B} H(p_{\theta}(\hat{x}_{i}))
\]  
(17)

3 Experimental Setup

Datasets We evaluate our methods on following two benchmarks for vision-language navigation in real 3D environments. **R2R** (Anderson et al., 2018) is a VLN dataset collected in photo-realistic environments (Matterport3D (Chang et al., 2017)). **RxR** (Ku et al., 2020) is a multilingual (English, Hindi, and Telugu) and larger than other existing VLN datasets with 11089, 232, 1517 and 2684 paths in train, val-seen, val-unseen, and test splits respectively. Please refer to Appendix A.3 for dataset details.

Evaluation Metrics On the R2R benchmark, we use standard metrics: Trajectory Length (TL), Navigation Error (NE ↓), Success Rate (SR ↑), and Success weighted by inverse Path Length (SPL ↑). On RxR benchmark, we use the standard metrics: SR ↑, SPL ↑, Coverage weighted by Length Score (CLS ↑ (Jain et al., 2019)), Normalized Dynamic Time Warping (nDTW ↑ (Ilharco et al., 2019)), and Success weighted by normalized Dynamic Time Warping (sDTW ↑ (Ilharco et al., 2019)).

Baselines We compare the single-run performance of different agents (see Table 1 and Table 2) on R2R and RxR. We conduct experiments over both benchmarks with our model-agnostic DAVIS added to baselines: 1) CLIP-ViL (Shen et al., 2021) is an extension of EnvDrop (Tan et al., 2019), which leverages the representation from CLIP (Radford et al., 2021) to replace the visual features extracting backbone from ResNet (He et al., 2016) to ViT (Dosovitskiy et al., 2021)). This baseline utilize back-translation as their test-time adaption. 2) VLN⊙BERT (Hong et al., 2021) proposes a recurrent BERT model that is time-aware for use in VLN. The parameters of our implemented baseline are initialized from PREVALENCE (Hao et al., 2020a). More details of the baselines can be found in Appendix A.4. And please refer to Appendix A.5 for the experimental implementation details with configuration illustrations.

Note that for each base architecture, we first imple-
Table 1: Model-agnostic Improvement over Powerful Baselines on R2R Benchmark. DAVIS is our full architecture applied over baselines. BEST and the SECOND best results are highlighted.

| Model                      | Validation Seen | Validation Unseen |
|----------------------------|-----------------|-------------------|
|                            | TL   | NE | SR | SPL | TL   | NE | SR | SPL |
| 0 Random                   | 9.58 | 9.45 | 16 | -   | 9.77 | 9.23 | 16 | -   |
| 1 Human                    | -    | -   | -  | -   | -    | -   | -  | -   |
| 2 Seq2Seq (Anderson et al., 2018) | 11.33 | 6.01 | 39 | -   | 8.39 | 7.81 | 22 | -   |
| 3 Speaker-Follower (Fried et al., 2018) | -    | 3.36 | 66 | -   | -    | 6.62 | 35 | -   |
| 4 PRESS (Li et al., 2019b) | 10.57 | 4.39 | 58 | 55  | 10.36 | 5.28 | 49 | 45  |
| 5 AuxRN (Zhu et al., 2020b) | -    | 3.33 | 70 | 67  | -    | 5.28 | 55 | 50  |
| 6 PREV ALENT (Hao et al., 2020a) | 10.32 | 3.67 | 69 | 65  | 10.19 | 4.71 | 58 | 53  |
| 7 RelGraph (Hong et al., 2020) | 10.13 | 3.47 | 67 | 65  | 9.99  | 4.73 | 57 | 53  |
| 8 CLIP-ViL (Shen et al., 2021) | 12.59 | 4.13 | 60.32 | 55.44 | 11.46 | 5.13 | 52.45 | 46.76 |
| 9 Nearest Neighbor Contrastive | 9.97 | 4.40 | 63.51 | 57.58 | 9.62 | 4.45 | 52.19 | 48.24 |
| 10 Test-time Adaptation (DAVIS) | 10.16 | 4.05 | 64.67 | 61.50 | 10.46 | 4.41 | 54.04 | 49.14 |
| 11 VLN-BERT (Hong et al., 2021) | 10.92 | 2.96 | 72.48 | 68.03 | 11.04 | 4.13 | 59.86 | 55.09 |
| 12 Nearest Neighbor Contrastive | 11.13 | 2.66 | 74.53 | 69.92 | 11.04 | 3.92 | 62.22 | 56.56 |
| 13 Test-time Adaptation (DAVIS) | 12.45 | 3.16 | 80.48 | 76.19 | 12.65 | 3.16 | 67.18 | 60.51 |

Table 2: Model-agnostic improvement over baselines on RxR Benchmark. DAVIS is our full architecture applied over baselines. BEST and the SECOND best results are highlighted.

| Model                       | Validation Seen | Validation Unseen |
|-----------------------------|-----------------|-------------------|
|                            | SR | SPL | CLS | nDTW | sDTW | SR | SPL | CLS | nDTW | sDTW |
| 1 EnvDrop (Tan et al., 2019) | -  | -   | -   | -    | -    | 38.5 | 34 | 54 | 51 | 32  |
| 2 Syntax (Li et al., 2021b) | -  | -   | -   | -    | -    | 39.2 | 35 | 56 | 52 | 32  |
| 3 CLIP-ViL (Shen et al., 2021) | 44.35 | 41.39 | 59.96 | 56.43 | 38.41 | 39.72 | 35.66 | 55.35 | 51.85 | 32.51 |
| 4 Nearest Neighbor Contrastive | 46.18 | 45.43 | 61.18 | 57.98 | 41.06 | 40.11 | 36.45 | 56.73 | 53.08 | 33.73 |
| 5 Test-time Adaptation (DAVIS) | 47.72 | 46.71 | 61.80 | 58.30 | 42.00 | 40.72 | 38.32 | 57.80 | 53.80 | 34.10 |
| 6 VLN-BERT (Hong et al., 2021) | 48.21 | 45.84 | 61.32 | 59.23 | 41.86 | 43.16 | 38.32 | 56.24 | 53.12 | 35.21 |
| 7 Nearest Neighbor Contrastive | 49.50 | 47.12 | 61.90 | 60.01 | 42.53 | 46.01 | 39.09 | 57.33 | 54.11 | 36.80 |
| 8 Test-time Adaptation (DAVIS) | 50.64 | 48.33 | 62.05 | 61.32 | 43.80 | 47.66 | 39.85 | 58.07 | 54.52 | 37.16 |

We report the evaluation performance of the proposed framework in a model-agnostic setting over powerful baselines, CLIP-ViL and VLN-BERT under Validation Seen and Validation Unseen splits.

As shown in Table 1, our base model initialized with VLN-BERT performs better than previous methods on the R2R benchmark. For CLIP-ViL, NNC achieve 3.86% and 3.17% improvement compared with CLIP-ViL baseline on SPL over Validation seen and unseen, respectively. With TTA, the model achieves further consistent improvement of 10.93% and 5.09% over baseline accordingly. This indicates the effectiveness of test-time adaptation. Similarly for VLN-BERT baseline, NNC achieves 2.78% and 2.67% improvement, TTA achieves 12.00% and 9.84% improvement on SPL over validation seen and unseen respectively. In general, for the metrics (TL, NE, SR, and SPL) that research recognizes for the R2R benchmark, our proposed DAVIS achieve consistent performance gain, which indicates effective generalization to the unseen environment. Results on the RxR benchmark are provided in Table 2. We consider SR, SPL, CLS, nDTW, and sDTW metrics on Validation seen and unseen split. For CLIP-ViL, with NNC, the model consistently improves 2.37% and 3.78% on nDTW and sDTW accordingly in Validation unseen. With TTA, the model achieves further consistent improvement of 3.76% and 4.89% accordingly. We also observe consistent performance gain of NNC over VLN-BERT with 1.86% and 4.51% on nDTW and sDTW respectively in Validation unseen. And the model updated with TTA achieves further consistent improvement of 2.64%
and 5.53% accordingly. DAVIS achieves consistent performance gain on the metrics. This further confirms model-agnostic effective generalization of DAVIS to the unseen environment.

4.2 Module Ablation

Effect of Test-time Adaptation To study the effect of test-time adaptation (TTA), we show results of both the full DAVIS model and Nearest Neighbor Contrastive (NNC) variant without TTA. As shown in Table 1 and Table 2, the test-time adaptation pushes the performance improvement even higher. For CLIP-ViL, TTA achieve 6.81% and 6.80% improvement over NNC on SR and SPL on Validation Seen, 3.54% and 1.87% on Validation Unseen. For VLN○BERT, TTA achieve 7.98% and 8.97% improvement over NNC on SR and SPL on Validation Seen, 7.98% and 6.98% on Validation Unseen. This indicates the test-time adaptation is irreplaceable for learning the generalization to the unseen environment.

Effect of Objectives We further study the effect of each component of our proposed framework over the R2R benchmark. To understand the contribution of each component, we split the objectives into three parts, $ML$, $CL_{IL}$, and $CL_{RL}$. Results are reported in Table 3. $ML$ represents the objective of the supervised part of DAVIS, a combination of Imitation Learning (IL) and Reinforcement Learning (RL), which is adopted from the common architecture applied in vision-and-language navigation. To ensure visual consistency, we have a contrastive learning objective of IL $CL_{IL}$ and a contrastive learning objective of RL $CL_{RL}$. On Validation unseen, the CLIP-ViL based variant without $CL_{IL}$ objective experience 6.60% and 3.42% drop on SR and SPL, respectively. The variant without $CL_{RL}$ objective experiences 5.63% and 3.42% drop on SR and SPL, respectively. We observe 16.62% and 15.81% drop of the VLN○BERT based variant without $CL_{IL}$ objective on SR and SPL, respectively. Without $CL_{RL}$ objective experience 13.16% and 16.35% drop on SR and SPL respectively. To study the performance of a pure self-supervised variant of DAVIS, we remove the $ML$ objective and train the model only with $CL_{IL}$ and $CL_{RL}$. On Validation Seen and Unseen, the self-supervised DAVIS based on CLIP-ViL and VLN○BERT experience performance drop, which indicates the importance of the supervised part.

Efficiency Analysis We keep the test-time augmentations at a reasonable number and achieve a balance between efficiency and accuracy. We show the inference time comparison on the R2R benchmark. For validation unseen split, CLIP-ViL and VLN○BERT baseline spent 23 seconds and 85 seconds. With test-time adaptation, the inference time increase to 70 and 190 seconds. For validation seen split, CLIP-ViL and VLN○BERT baseline spent 23 seconds and 40 seconds. With test-time adaptation, the inference time increase to 85 and 89 seconds. The increased time is mainly consumed by the extra adaptation and augmentations. When applying selective augmentations and test-time adaptation, the inference time is 41, 34 for CLIP-ViL based and 57, 103 for VLN○BERT on seen and unseen, respectively. This indicates the balance between efficiency and accuracy.

4.3 Qualitative Analysis

In Figure 3, we visualize the navigation trajectory of baseline and DAVIS with the architecture of VLN○BERT under R2R unseen environment. We show the panoramic view of the start point, intermediate steps, and stop point. Given the same instruction and the start point view, the visualization of trajectory predicted by both models demonstrates the generalization superiority of our DAVIS.

| Model       | $ML$ | $CL_{IL}$ | $CL_{RL}$ | Validation Seen | Validation Unseen |
|-------------|------|-----------|-----------|-----------------|-------------------|
|             | TL   | NE ↓  | SR ↑  | SPL ↑ | TL   | NE ↓  | SR ↑  | SPL ↑ |
| 1 CLIP-ViL  | ✓    | ✓      | ✓      | ✓     | 10.66 | 4.33  | 63.83 | 52.76 |
| 2 CLIP-ViL  | ✓    | ✓      | ✓      | ✓     | 10.34 | 4.27  | 62.15 | 50.05 |
| 3 CLIP-ViL  | ✓    | ✓      | ✓      | ✓     | 11.36 | 5.36  | 47.48 | 42.57 |
| 4 VLN○BERT | ✓    | ✓      | ✓      | ✓     | 11.55 | 3.92  | 68.85 | 54.74 |
| 5 VLN○BERT | ✓    | ✓      | ✓      | ✓     | 11.52 | 4.10  | 69.35 | 56.46 |
| 6 VLN○BERT | ✓    | ✓      | ✓      | ✓     | 12.70 | 4.20  | 57.59 | 51.89 |

Table 3: Ablation on R2R Benchmark. $ML$, $CL_{IL}$ and $CL_{RL}$ represent supervised loss, contrastive loss of imitation learning and reinforcement learning respectively. Full model (DAVIS) in Table 1 consists of all three losses. BEST and the SECOND best results are highlighted.
Walk through the double doors into the building and turn left. Walk through the hallway and turn left. Stop just inside the door.

(a) Succeed: Trajectory predicted by our TTC.
(b) Failed: Trajectory predicted by RecBERT.

Figure 3: Navigation Trajectory Visualization. The panoramic view of the start point, intermediate steps and the stop point of the predicted trajectories by VLN⟲BERT and DA VIS are visualized.

VLN⟲BERT failed to recognize the double door of the building and make wrong action prediction to the next entrance. It made the wrong inference of the hallway and stops at the wrong position. While our DAVIS successfully refer to the double doors and follow the instruction inside the building, and turn left at the correct viewpoint. Thus select the right hallway to walk through and turn left. Finally, the agent stops just at the targeted position. More trajectory visualization and the bird view comparisons can be found in Appendix A.6.

5 Related Work

Vision-language Navigation (Wang et al., 2019a; Tan et al., 2019; Shen et al., 2021; Hong et al., 2021) learn to ground cross-modal reasoning challenge via imitation learning and reinforcement learning. Prior studies (Fu et al., 2020; Parvaneh et al., 2020; Liu et al., 2021; Majumdar et al., 2020) achieve improvement of the generalizability by proposing diversified forms of augmentations. Compared with the most related work, Wang et al. (2019b) adapt model in all the samples, and adapt in the house-level. In contrast, we adopt a more realistic setting that adapt in one single sample without any assumptions of the house information, which further improves the generalization for the VLN agent. Besides, DAVIS is model-agnostic to either the sequence-to-sequence or transformer-based architectures.

Test-time Adaptation Test-time adaptation aims at leveraging test assumptions to address the problem of distribution shift. Most of the previous studies can be divided into two categories, test-time training and test-time augmentation. Test-time training (TTT) (Sun et al., 2020) proposes a model that composes of both the supervised module and self-supervised module and enables a different training procedure on test inputs. Test-time augmentation (TTA) has been proved effective in addressing the problem of distribution shift and achieving generalization to test set. We propose to combine the advantages of both TTT and TTA regime with a two-stage semi-supervised model design with considerations of maximizing agreements over augmentations to adapt to distribution shift at test-time.

6 Conclusion

We propose a Test-time Visual Consistency (DAVIS) framework that improves the generalizability of the VLN agents to unseen environments. DAVIS is composed of a supervised branch and a self-supervised branch based on Imitation Learning and Reinforcement Learning. In addition to standard semi-supervised joint training procedure, DAVIS implements test-time adaptation before inference. Experimental results on R2R and RxR benchmarks confirm the superiority of DAVIS over previous state-of-the-art VLN baselines.
Limitations

In general, the limitations of our work are from three aspects. First, the visual backgrounds in the dataset we are using are from the English-Speaking country. Though the multi-lingual instructions are provided, they are still using the English instructions as pivot and thus biased to the English language navigation habits. Second, we focus on the in-door environment with sufficient data and may result in limited performance in the low-resource situation. Finally, how to scale up the framework to multi domain settings remain underexplored. In the future work, we hope to solve the issues together in a simple unified setting and bring the model to broader VLN applications.

Ethics Statement

Since the VLN challenge is a fundamental problem in the field of vision and language, we do not foresee any significant ethical issues. Minor concerns are the biases introduced from the experimental dataset and baseline model side. In fact, the dataset we used (R2R, RxR), and the baseline model we used (see section 3) are all published. Thus these concerns seem low-risk for our experimental evaluations. In addition, we did not notice any such problems in our work.

References

Abulkem Abuduweili, Xingjian Li, Humphrey Shi, Chengzhong Xu, and Dejing Dou. 2021. Adaptive consistency regularization for semi-supervised transfer learning. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6919–6928.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3674–3683.

Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. 2017 International Conference on 3D Vision (3DV), pages 667–676.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv, abs/2010.11929.

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.

Tsu-Jui Fu, Xin Eric Wang, Matthew F Peterson, Scott T Grafton, Miguel P Eckstein, and William Yang Wang. 2020. Counterfactual vision-and-language navigation via adversarial path sampler. In European Conference on Computer Vision, pages 71–86. Springer.

Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B. Tenenbaum. 2020. Look, listen, and act: Towards audio-visual embodied navigation.

Tuomas Haarnoja, Aurick Zhou, P. Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In ICML.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. 2020a. Towards learning a generic agent for vision-and-language navigation via pre-training. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13134–13143.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. 2020b. Towards learning a generic agent for vision-and-language navigation via pre-training.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9726–9735.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778.

David S. Hippocampus. 2020. A simple imitation learning method via contrastive regularization.

Yicong Hong, Cristian Rodriguez-Opazo, Yuankai Qi, Qi Wu, and Stephen Gould. 2020. Language and visual entity relationship graph for agent navigation. ArXiv, abs/2010.09304.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. Vlnbert: A recurrent vision-and-language bert for navigation. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1643–1653.
Ronghang Hu, Daniel Fried, Anna Rohrbach, Dan Klein, Trevor Darrell, and Kate Saenko. 2019. Are you looking? grounding to multiple modalities in vision-and-language navigation.

Sangeek Hyun, Jihwan Kim, and Jae-Pil Heo. 2021. Self-supervised video gans: Learning for appearance consistency and motion coherency. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10821–10830.

Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. ArXiv, abs/1907.05446.

Vihan Jain, Gabriel Magalhães, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. Stay on the path: Instruction fidelity in vision-and-language navigation. ArXiv, abs/1905.12255.

Minki Jeong, Seokeon Choi, and Changick Kim. 2021. Few-shot open-set recognition by transformation consistency. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12561–12570.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4392–4412.

Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang. 2021a. 3d human action representation learning via cross-view consistency pursuit. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4739–4748.

Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2021b. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. ArXiv, abs/2012.15409.

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

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

Chong Liu, Fengda Zhu, Xiaojun Chang, Xiaodan Liang, and Yi-Dong Shen. 2021. Vision-language navigation with random environmental mixup. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 1624–1634.

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. ArXiv, abs/2004.14973.

Luke Melas-Kyriazi and Arjun Manrai. 2021. Pix-match: Unsupervised domain adaptation via pixel-wise consistency training.

Yassine Ouali, Céline Hudelot, and Myriam Tami. 2020. Semi-supervised semantic segmentation with cross-consistency training. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12671–12681.

Amin Parvaneh, Ehsan Abbaspour, Damien Teney, Qinfeng Shi, and Anton van den Hengel. 2020. Counterfactual vision-and-language navigation: Unravelling the unseen. In NeurIPS.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In ICML.

Chao Ren, Xiaohai He, Chuncheng Wang, and Zhibo Zhao. 2021. Adaptive consistency prior based deep network for image denoising. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8592–8602.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2021. How much can clip benefit vision-and-language tasks? arXiv preprint arXiv:2107.06383.

A. Srinivas, Michael Laskin, and P. Abbeel. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. In ICML.

Yu Sun, X. Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In ICML.

Hao Tan, Licheng Yu, and Mohit Bansal. 2019. Learning to navigate unseen environments: Back translation with environmental dropout. ArXiv, abs/1904.01959.

Xin Wang, Qiyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan Fang Wang, William Yang Wang, and Lei Zhang. 2019a. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June:6622–6631.

Xin Wang, Qiyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019b. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation.
Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. 2021. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 16679–16688.

Wending Yan, Robby T. Tan, Wenhao Yang, and Dengxin Dai. 2021. Self-aligned video deraining with transmission-depth consistency. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11961–11971.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. 2020a. Vision-language navigation with self-supervised auxiliary reasoning tasks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10009–10019.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. 2020b. Vision-language navigation with self-supervised auxiliary reasoning tasks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10009–10019.

A Appendix

A.1 Method Details

As shown in Figure 4(a), the policy $\psi_{CL}$ network learn to maximize the agreement of action decisions $a_t$ and $\hat{a}_t$ over positive observation pairs. As shown in Figure 4(b), we consider taking augmentations for introducing consistency during actor critic learning.

A.2 Background

Vision-language Navigation (Fu et al., 2020; Parvaneh et al., 2020) utilize the counterfactuals to augment either the trajectories or the visual observations. (Liu et al., 2021) proposes to mixup the environment. (Majumdar et al., 2020) leverages large-scale image-text pairs for the Web Data. By training on a large amount of image-text-action triplets in a self-supervised learning manner, the pre-trained model provides generic representations of visual environments and language instructions. Some studies (Hu et al., 2019; Gan et al., 2020) learn to utilize other available modalities. (Hao et al., 2020b) presents pre-training and fine-tuning paradigm for vision-and-language navigation (VLN) tasks. (Wang et al., 2019b) propose a novel Reinforced Cross-Modal Matching (RCM) approach that enforces cross-modal grounding both locally and globally via reinforcement learning (RL) to address three critical challenges for this task: the cross-modal grounding, the ill-posed feedback, and the generalization problems.

Consistency Regularization (Xie et al., 2021; Melas-Kyriazi and Manrai, 2021) learn dense feature representations via pixel-level visual consistency. (Ouali et al., 2020; Abuduweili et al., 2021; Jeong et al., 2021) learn consistency regularization for semantic segmentation. (Ren et al., 2021) proposes an Adaptive Consistency Prior based Deep Network for Image Denoising. Recent studies (Li et al., 2021a; Hyun et al., 2021; Yan et al., 2021) show increasing interest of encouraging consistency across multiple modalities. (Srinivas et al., 2020) learns contrastive unsupervised representations for reinforcement learning. (Hippocampus, 2020) proposes to train the policy as a classifier via contrastive regularization for imitation learning. In this study, we propose to leverage consistency into both imitation learning and reinforcement learning process of the VLN agent. Specifically, we utilize momentum contrast for enforcing visual consistency between query panoramic features as well
as stable decision makings across similar observations.

A.3 Dataset Details
The details of our experimental datasets are:

- **R2R** (Anderson et al., 2018): is a VLN dataset collected in photo-realistic environments (Matterport3D (Chang et al., 2017)). It contains 61, 11 and 18 scenes for training, validation and testing, respectively.

- **RxR** (Ku et al., 2020): is a multilingual (English, Hindi, and Telugu) and larger (more extended instructions and trajectories) than other existing VLN datasets. It contains 11089, 232, 1517 and 2684 paths in train, val-seen (train environments), val-unseen (val environments), and test splits respectively.

A.4 Baseline Details
We compare the single-run performance of different agents including Seq2Seq (Anderson et al., 2018), Speaker-Follower (Fried et al., 2018), PRESS (Li et al., 2019b), AuxRN (Zhu et al., 2020b), PREVALENT (Hao et al., 2020a) and RelGraph (Hong et al., 2020) on R2R, EnvDrop (Tan et al., 2019) and Syntax (Li et al., 2021b) on RxR. We conduct experiments over both benchmarks with our model-agnostic DAVIS added to baselines:

1) **CLIP-VIL** (Shen et al., 2021) is an extension of EnvDrop (Tan et al., 2019), which leverages the representation from large-pretrained CLIP (Radford et al., 2021) model to replace the visual features extracting backbone from ResNet (He et al., 2016) to Vision Transformer (ViT (Dosovitskiy et al., 2021)).

2) **VLNBERT** (Hong et al., 2021) proposes a recurrent BERT model that is time-aware for use in VLN. The parameters of our implemented baseline are initialized from PREVALENT (Hao et al., 2020a).

- **CLIP-VIL** (Shen et al., 2021): is an extension of EnvDrop (Tan et al., 2019), which leverages the representation from large-pretrained CLIP (Radford et al., 2021) model to replace the visual features extracting backbone from ResNet (He et al., 2016) to Vision Transformer (ViT (Dosovitskiy et al., 2021)).
VLNBERT (Hong et al., 2021): proposes a recurrent BERT model that is time-aware for use in VLN. The parameters of our implemented baseline are initialized from PREVALENT (Hao et al., 2020a).

A.5 Implementation Details

There are $K=36$ view images in each panoramic observation with available panoramic action space. The observation features are ResNet image features for VLN$\bowtie$BERT and ViT image features for CLIP-ViL. The language encoder is LSTM for sequence-to-sequence architecture and BERT for transformer architecture. The architecture consists of a Speaker and a Listener following the implementation in EnvDrop. The speaker is either a sequence-to-sequence or a transformer-based module to estimate the likelihood of instruction for the given trajectory. The listener accordingly follows the instruction from the speaker that estimates the likelihood of action sequence for the instruction-trajectory pair. We use Adam Optimizer and batch sizes 32 for training and test-time training. The learning rates are set as $1e-4$. We train 20000 iterations for the training stage and 10 iterations for the test-time stage. We set the balance factors $\lambda_{ml}$, $\lambda_{rl}$, $\lambda_{dl,ml}$, $\lambda_{dl,rl}$ as 0.2, 0.2, 0.2 and 0.2 by grid search respectively. The parameter size of our DAVIS based on CLIP-ViL and VLN$\bowtie$BERT is 91M and 160M respectively.

Computation and Resources We use four single NVIDIA A100 GPU Server for all the experiments. The computation is used for the training and testing stage. The average runtime for our proposed DAVIS is discussed in Section 4.2.

A.6 Qualitative Results

To further look at the high-level decision-making process of the agent, we visualize the bird view of trajectories predicted by the baseline and DAVIS, compared with ground truth trajectory in Figure 6. We randomly select one case for Scene A and two cases for Scene B under the R2R validation unseen environment. For Case 1 under Scene A, the trajectory predicted by DAVIS matches the Shortest Path. Meanwhile, the VLN$\bowtie$BERT baseline heads the wrong way initially and ends up with redundant trajectories. Similar to Case 2 and Case 3 under Scene B, the baseline is more likely to miss the key reference in the new environment and head in the wrong direction. In contrast, our proposed DAVIS shows incredible generalization under unseen environments and follows the instruction continually. We also showcase a trajectory visualization in Figure 7.
Figure 6: **Bird View Comparison.** The green line represents the trajectory of ground truth (Shortest Path). The red line and the yellow line represent the trajectory of VLN\(\bigcirc\)BERT baseline and VLN\(\bigcirc\)BERT based DAVIS respectively. The trajectory points from the start position to the end position. The leftmost is Scene A, and the rightmost two scenes are Scene B.

Turn left and walk down the hallway. When you reach the black chair at the wall, turn left. Once you turn left walk forward and enter the next room on your left. Stop once you are behind the first yellow chair.

Figure 7: Trajectory Visualization.