Learning Disentanglement with Decoupled Labels for Vision-Language Navigation

Supplementary Material

Wenhao Cheng[1†][0000−0003−4108−1647], Xingping Dong[2†][0000−0003−1613−9288], Salman Khan[3], and Jianbing Shen[4⋆][0000−0003−1883−2086]

[1] School of Computer Science, Beijing Institute of Technology
[2] Inception Institute of Artificial Intelligence, UAE
[3] Mohamed bin Zayed University of Artificial Intelligence, UAE
[4] SKL-IOTSC, Computer and Information Science, University of Macau

A Model Details

A.1 VLN↻BERT

As discussed in Section 3.2, VLN↻BERT [9] encodes the language instruction once at the beginning. The leading input token is selected to initialize the agent's state. During navigation, there is no language processing, but the state will be refined by both the attended language and visual features. Next, we present the refinement in detail.

At each navigation step t, the prior state h_{t-1} will be firstly fused with previous action embedding a_{t-1} to encode the history information. Formally:

$$\bar{h}_t = \text{LayerNorm}([h_{t-1}; a_{t-1}]W_a), \tag{1}$$

where LayerNorm denotes layer normalization [2], $[\cdot]$ represents concatenation operation, and W_a is trainable parameter.

To enrich the state representation, the matched language and visual features are fed to the current state. Specifically, the weighted sum of language features is calculated as $F^l = \alpha(I_t)I$, where I is the encoded instruction, and $\alpha(I_t)$ means the attention scores over the language tokes at the final encoding layer, normalized by a Softmax function.

In terms of the attended visual scene features, different from the baseline, we use the weighted sum of the landmark- and action-aware outputs in disentangled decoding module as the attention scores $\alpha(V_t)$ over the visual tokens, where V_t is the visual scene feature at each candidate direction. Then the weighted visual features are expressed as $F^v = \alpha(V_t)V_t$.

Finally, the state is refined as:

$$h_t = \text{LayerNorm}([\hat{h}_t; F^v \odot F^l]W_r), \tag{2}$$

⋆ Corresponding author: *Jianbing Shen* (shenjianbingcg@gmail.com). † Equal contribution.

where $\hat{h}_t$ is the output state representation of the final encoding layer, $\odot$ denotes the element-wise product, and W_r is trainable parameter.

A.2 HAMT

HAMT [4] is a multi-modal transformer that integrates all past historical information in VLN. Next, we present the history encoding in detail. The panoramic observation is processed via a hierarchical vision transformer ViT [5]. Firstly, each oriented view image feature $v_{t,i}$ at each step t is encoded by:

$$v_{t,i} = \text{LayerNorm}(W_f f_{t,i}) + \text{LayerNorm}(W_o o_{t,i}), \tag{3}$$

where $f_{t,i}$ is image feature at direction $o_{t,i}$ in view i, embedded via a ViT, W_f and W_o are learnable parameters. Then, another panoramic transformer is enforced to learn spatial relationship among all directions. To save computation cost, the observation at each step is integrated into one temporal vector v_t^h by average pooling. The final temporal token h_t is calculated by:

$$h_t = \text{LayerNorm}(W_h v_t^h) + \text{LayerNorm}(W_a a_{t-1}^h), \tag{4}$$

where a_{t-1}^h denotes the relative angle representation of previous action. For the proposed DDL, we keep the historical encoding unchanged, and only disentangle the input observation at the current step.

A.3 OAAM

OAAM [11] utilizes two learnable attention modules to highlight the corresponding object- and action-related part of the given instruction, and then combines the predictions as the final decision. Next, we discuss the navigation process in detail. The internal memory is maintained by a LSTM [8]:

$$h_t = \text{LSTM\_Decoder}([\hat{f}_t; a_{t-1}], \hat{h}_{t-1}) \tag{5}$$

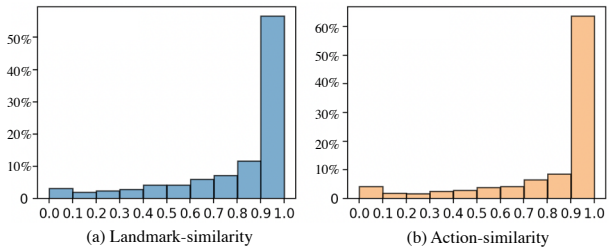
where a_{t-1} is the previous action embedding, $\hat{f}_t$ is the current observation, and $\hat{h}_t = \text{Tanh}(W_l[h_t; \hat{u}_t])$ is instruction-aware hidden state, where $\hat{u}_t$ is the attentive instruction feature. To obtain the object-aware hidden state $\hat{h}_t^o$, the first attention module is applied:

$$\gamma_{t,j} = \text{Softmax}_j(u_j^T W_o h_t), \quad \hat{u}_t^o = \sum_j \gamma_{t,j} u_j, \quad \hat{h}_t^o = \text{Tanh}(W_p[h_t; \hat{u}_t^o]), \tag{6}$$

where u_j is the j-th word embedding in instruction, $\gamma_{t,j}$ is object-aware language attention weight, and W_p, W_o are trainable parameters.

Similarly, the action-aware hidden state $\hat{h}_t^a$ and language attention weight $\sigma_{t,j}$ can be obtained. Then the intermediate action confidence is formulated by:

$$G^{OA}(a_{t,k}) = f_{t,k}^T W_r \hat{h}_t^o, \quad G^{AA}(a_{t,k}) = o_{t,k}^T W_t \hat{h}_t^a, \tag{7}$$



Supp-Figure 1. Cosine similarity distribution between annotated labels and pseudo-labels at all viewpoints on the validation unseen set of LAR2R.

where $f_{t,k}$ is visual feature and $o_{t,k}$ is orientation feature of each candidate direction. The final action probability of each navigable view k is determined by the weighted sum of the object-aware confidence $G^{OA}(a_{t,k})$ and action-aware confidence $G^{AA}(a_{t,k})$. The combined weight is formulated as: $w_u = W_d \hat{u}_t$, which is a weighted vector to measure the importance of the object- and action-aware instruction at each viewpoint and W_d is trainable parameter.

The model is trained by mixed Imitation Learning (IL) and Reinforcement Learning (RL). There are three ways of action selection for different scenarios. In IL, the agent follows teacher action a_t^*. In RL, the selected action is sampled with probability distribution $a_t \sim P(a_{t,k})$. During testing, the agent chooses the candidate with maximal probability: $a_t = \text{argmax}_k P(a_{t,k})$.

B Dataset

We evaluate our method on two datasets R2R [1] and R4R [10]. The R2R is built upon Matterport3D Simulator [3] with photo-realistic environment. The original R2R dataset consists of 7,189 paths in 90 real-world scenes. Each path has 3 or 4 instructions generated by human annotation, with an average of 29 words per instruction and 10m of physical length per path ranging from 5 to 7 viewpoints. The dataset is split into four sets: *training*, *validation seen*, *validation unseen* and *test unseen*, where *unseen* means that paths are sampled from environments that are not seen during training. Specifically, 4,676 paths are used for training and 340 paths for validation seen in 61 scenes, 783 paths for validation unseen in 11 scenes, and the remaining 18 scenes with 1,391 paths for testing. The R4R extends R2R with longer instructions and paths. The R4R contains three splits: *training* (233,613 instructions), *validation seen* (1,035 instructions) and *validation unseen* (45,162 instructions).

C Additional Ablations

As discussed in Section 3.4, in disentangled decoding module, we firstly equip the disentanglement branch to the original VLN⟳BERT, then the decoupled labels are utilized to optimize the language attention weight via the proposed language auxiliary loss. The VLN⟳BERT is initialized by a pre-trained model. Here, we

Component			R2R Val seen				R2R Val unseen			
Base	DB	DL	SR↑	SPL↑	OR↑	NE↓	SR↑	SPL↑	OR↑	NE↓
✓			71.5	67.4	76.8	2.87	62.2	56.5	68.3	4.09
✓	✓		74.2	68.8	81.0	2.72	63.1	57.4	70.5	3.91
✓	✓	✓	**77.5**	**71.4**	**83.9**	**2.47**	**64.8**	**58.3**	**71.1**	**3.84**

Supp-Table 1. Ablation study showing the effect of different components in VLN⟳BERT on R2R. DB means the disentanglement branch, and DL represents the decoupled labels.

choose PREVALENT [7] to obtain state-of-the-art performance. Thus, the training consists of one stage, *i.e.*, directly train the model with mixed original data and augmented data. Supp-Table 1 shows the ablations of each component. We can find when the baseline is equipped with the disentanglement branch, it has obtained performance improvement. This indicates that disentangling the visual and orientation features can benefit the agent's understanding of complex input. Moreover, the decoupled labels can help the agent locate the specific positions of the landmark- and action-related parts in instruction, further improving the decision-making ability.

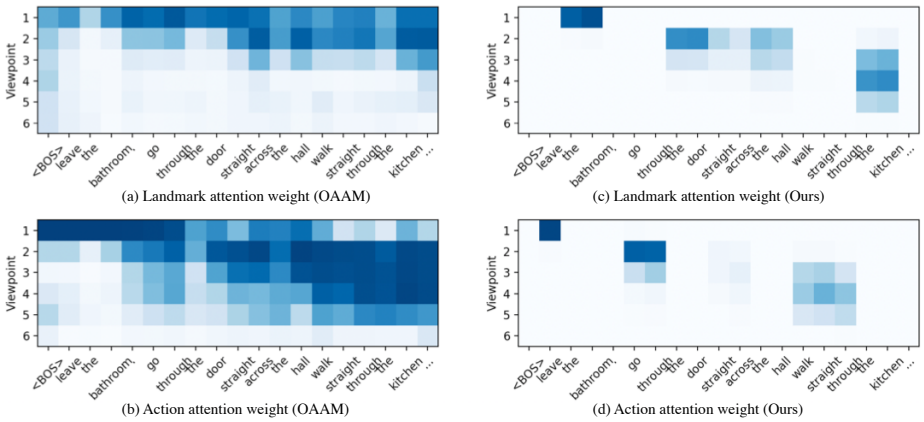
D Qualitative Results

Supp-Figure 2 shows the comparison of disentangled attention weights between OAAM and ours, indicating the disentanglement of our model is more accurate. Supp-Figure 3 also presents the visualization of a trajectory predicted by our Transformer-based model. The language attention weight concerning for the agent's state is disentangled clearly along the path, making the navigation more interpretable. We also provide a failure case in Supp-Figure 4. This shows object recognition needs to be improved in future work.

E Future Work

To learn this fine-grained scenario more efficiently, in future work, it would be necessary to explore how to accelerate the generation of labeled data via semi-/weakly- supervised methods. One direction is to enhance the ability of our decoupled label speaker and try to generate more accurate pseudo labels. In such a way, high-quality data annotations can be obtained on new more VLN datasets. Another way is to synthesize new instructions according to the annotated index in a weak manner. Since each trajectory usually has three instructions, cross-combination between different sub-instructions can generate more labeled data.

We also provide more discussion and insights for introducing few-shot learning to the VLN task in the future work. The VLN task requires the agent to navigate



(a) Landmark attention weight (OAAM)

(c) Landmark attention weight (Ours)

(b) Action attention weight (OAAM)

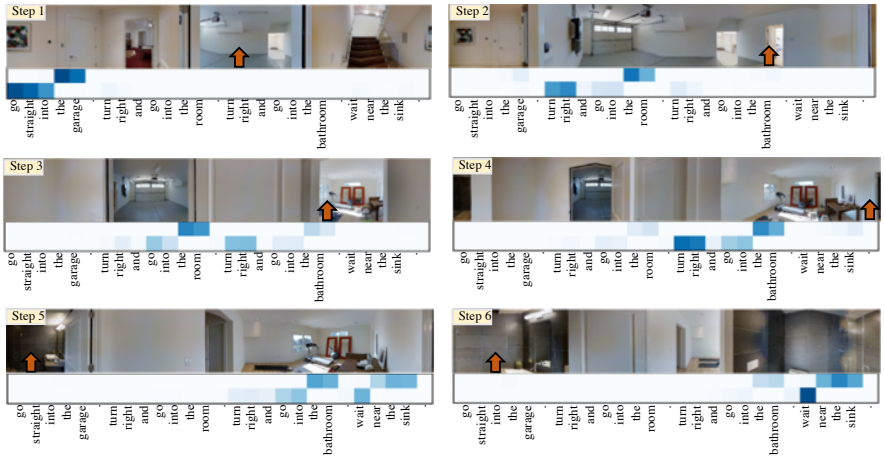
(d) Action attention weight (Ours)

Supp-Figure 2. A comparison of language attention weight predicted by the baseline OAAM [11] (left column) and our full model (right column). The disentanglement of our model is more accurate. Take the first viewpoint as an example. Our model is able to disentangle the "leave" and "the bathroom" more clearly while OAAM only tends to focus on the beginning of instruction with too much noise.

in an unseen environment, which is intrinsically similar to few-shot learning task that aims to learn a new function from limited annotation. The internal state during navigation is often maintained in a single-vector-based, history-bank-based, or graph-based manner. So if we only have a limited amount of labeled data, how to efficiently represent and mine the pivotal information during exploration? From the perspective of meta Reinforcement Learning (RL), fast adaption can be accelerated via multiple slow explorations [6]. At each training iteration, the agent could be allowed to explore a series of episodes, while integrating the time order information into the internal state. We believe that this will help the agent extract the meta knowledge and generalize to new unseen environments.

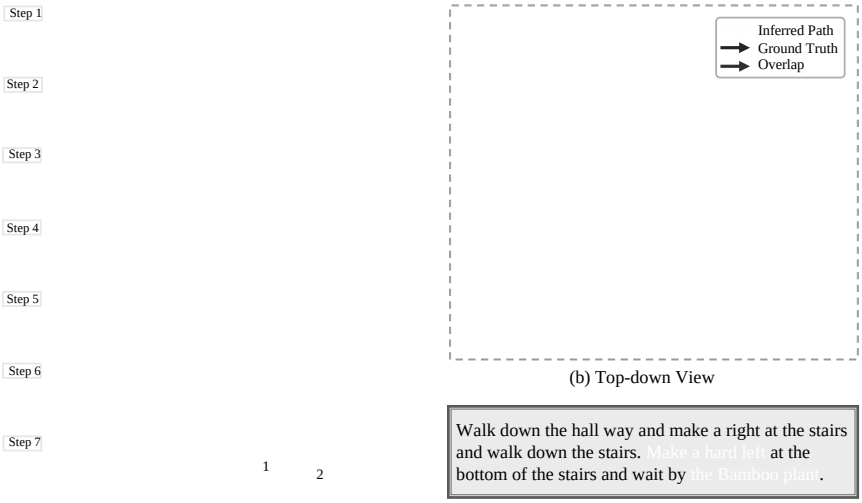
References

1. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3674–3683 (2018) 3
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 1
3. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: 7th IEEE International Conference on 3D Vision, 3DV 2017. pp. 667–676. Institute of Electrical and Electronics Engineers Inc. (2018) 3



Supp-Figure 3. Visualization of a trajectory with landmark (top row) and action (bottom row) attention weights of our model based on VLN⟳BERT [9]. We presented the averaged attention weight of state-language over all heads at each step. The image is depicted in panoramic view, and the arrow roughly denotes the agent's heading direction.

4. Chen, S., Guhur, P.L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. Advances in Neural Information Processing Systems **34** (2021) 2
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020) 2
6. Duan, Y., Schulman, J., Chen, X., Bartlett, P.L., Sutskever, I., Abbeel, P.: Rl2: Fast reinforcement learning via slow reinforcement learning. arXiv preprint arXiv:1611.02779 (2016) 5
7. Hao, W., Li, C., Li, X., Carin, L., Gao, J.: Towards learning a generic agent for vision-and-language navigation via pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13137–13146 (2020) 4
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997) 2
9. Hong, Y., Wu, Q., Qi, Y., Rodriguez-Opazo, C., Gould, S.: Vln bert: A recurrent vision-and-language bert for navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1643–1653 (June 2021) 1, 6, 7
10. Jain, V., Magalhaes, G., Ku, A., Vaswani, A., Ie, E., Baldridge, J.: Stay on the path: Instruction fidelity in vision-and-language navigation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 1862–1872 (2019) 3
11. Qi, Y., Pan, Z., Zhang, S., van den Hengel, A., Wu, Q.: Object-and-action aware model for visual language navigation. In: Proceedings of the European Conference on Computer Vision, Glasgow, Scotland. pp. 23–28. Springer (2020) 2, 5



(a) Panoramic View (b) Top-down View (c) Instruction

Supp-Figure 4. A failed case of our model based on VLN⟳BERT [9]. (a) is the predicted trajectory along each step. (b) is a comparison between ground truth and inferred path in a top-down view. (c) is the given instruction. The target location is marked as position 1 in Figure (a), but the agent stopped at position 2. This can be explained as the agent has a poor understanding of "the hard left" and "the Bamboo plant". Specifically, the first two steps of the agent are different from the ground truth, but it can be considered to follow the instruction (*Walk down the hall way and make a right at the stairs*). In the third step, the agent finds the stair and returns to the correct path. At the end, the agent turns left, but it does not match the *hard left*. Most importantly, the agent doesn't know *the Bamboo plant*, which caused the wrong prediction. Therefore, object recognition is a direction that needs to be explored in future work.