Supplementary material to “Are You Looking? Grounding to Multiple Modalities in Vision-and-Language Navigation”

A Details on the compared VLN models

The Speaker-Follower (SF) model (Fried et al., 2018b) and the Self-Monitoring (SM) model (Ma et al., 2019) which we analyze both use sequence-to-sequence model (Cho et al., 2014) with attention (Bahdanau et al., 2015) as their base instruction-following agent. Both use an encoder LSTM (Hochreiter and Schmidhuber, 1997) to represent the instruction text, and a decoder LSTM to predict actions sequentially. At each timestep, the decoder LSTM conditions on the action previously taken, a representation of the visual context at the agent’s current location, and an attended representation of the encoded instruction.

While at a high level these models are similar (at least in terms of the base sequence-to-sequence models – both papers additionally develop techniques to select routes from these base models during search-based inference techniques, either using a separate language generation model in SF, or a progress-monitor in SM), they differ in the mechanism by which they combine representations of the text instruction and visual input. The SM uses a co-grounded attention mechanism, where both the visual attention on image features and the textual attention on the instruction words are generated based on previous decoder LSTM hidden state h_{t-1}, and then the attended visual and textual features are used as LSTM inputs to produce h_t. The SF model only uses attended visual features as LSTM inputs and then produces textual attention based on updated LSTM state h_t. Also, the visual attention weights are calculated with an MLP and batch-normalization in SM, while only a linear dot-product visual attention is used in SF. Empirically these differences produce large performance improvements for the SM model, which may contribute to the smaller gap between the SM model and its non-visual counterparts.

B Details on the training mechanisms

Anderson et al. (2018) compare two methods for training agents, which subsequent work on VLN has also used. These methods differ in whether they allow the agent to visit viewpoints which are not part of the true routes at training time. In the first training setup, *teacher-forcing*, the agent visits each viewpoint in a given true route in sequence, and is supervised at each viewpoint with the action necessary to reach the next viewpoint in the true route. In the second training setup, *student-forcing*, the agent takes actions by sampling from its predicted distribution at each timestep, which results in exploring viewpoints that are not part of the true routes. At each viewpoint, supervision is provided by an oracle that returns the action which would take the agent along the shortest path to the goal. Empirically, student-forcing works better in nearly all settings in Table 1 (except for the non-visual version of the SF model), which is likely due to the fact that it reduces the discrepancy between training and testing, since it allows the agent to sample from its own prediction during training. Teacher-forcing works better for the non-visual version of the SF model, and we hypothesize that following the ground-truth routes during training allows the SF model to better preserve the geometric structures of the routes and match them to the instructions for the non-visual setting.

C Details on the object representation

In our object representation, we use the top-150 detected objects (with the highest detection confidence) at each location in the environment. The detection results are obtained from a Faster R-CNN detector (Ren et al., 2015) pretrained on the Visual Genome dataset (Krishna et al., 2017).