7 Supplementary

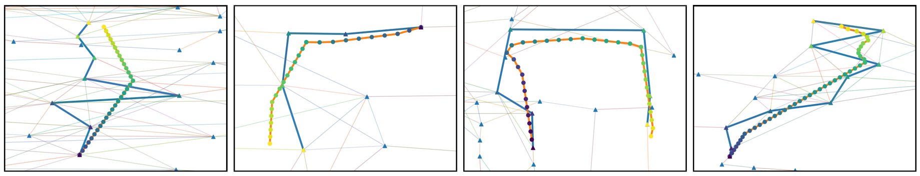

Fig. 5. Example conversions between VLN-CE paths (circles on orange line) and VLN nav-graph based trajectories (triangles on blue line). For both, darker markers are earlier in the trajectories. Other nav-graph nodes / edges are also shown. Note that the nav-graph is often a poor proxy for the 3D space with our agent paths in continuous space requiring zig-zag patterns in the nav-graph.

7.1 Converting VLN-CE Paths to Nav-Graph-based VLN

As discussed in Sec. 5.3, we convert paths from our agents in continuous environments to nav-graph trajectories for comparison with VLN. To do so, we apply a simple algorithm. Consider a trajectory in VLN-CE consisting of a sequence of positions $p_0, p_1, p_2, \ldots, p_T$. Note that the initial position p_0 aligns with the start node v_s for this trajectory from the VLN nav-graph. The goal then is to find a path through the nav-graph from this node that follows the continuous path.

Starting from v_s, we iteratively snap to the nearest adjacent node by minimizing distance from the current position. More concretely, at the beginning of the sequence we set the current node c to be v_s and consider a 'navigable set' $\mathcal{N}(c)$ consisting of all adjacent nodes to c as well as c. We then compute the distance between every node in $\mathcal{N}(c)$ and the continuous environment path position p_1. Whichever node from the active set is nearest to p_1 becomes the new current node. This is repeated for $p_2, p_3, \ldots, p_T$ with the current node (and thus navigable set) shifting to whichever node is nearest and within 1-step of the current node.

Fig. 5 shows some of the resulting trajectories (triangles on thick blue line) as well as the underlying continuous path (dots on orange line). These images also show the nav-graph with thin multi-colored lines for edges and blue triangles for panoramic nodes. Notably, the nav-graph comes up short in representing the continuous paths. Often our agents will navigate through spaces not captured by nav-graph nodes; resulting in nav-graph trajectories that have high error or must oscillate to follow the continuous path.

7.2 Is SPL still admissible?

The ground-truth trajectories in the R2R dataset are shortest paths, however, they are shortest paths on the nav-graph. Thus, these ground-truth trajectories need not be the shortest path (or even close to the shortest path) in the continuous space of VLN-CE. Such a discrepancy could lead to an agent that better follows the provided instruction – thereby going through all the waypoints the instruction directs it through – achieving a lower SPL than one that does not. To examine this, we compute the SPL of an oracle that takes the shortest path but must also travel through the waypoints. This oracle achieves 0.968 SPL across all paths in VLN-CE and 0.959 SPL in val-unseen, confirming that SPL is still valid for evaluation and to optimize while training.